# Red Hat Integration 2023.q2

## Release Notes for Red Hat Integration 2023.q2

What's new in Red Hat Integration

Last Updated: 2023-09-22

# Red Hat Integration 2023.q2 Release Notes for Red Hat Integration 2023.q2

What's new in Red Hat Integration

## Legal Notice

## Abstract

Describes the Red Hat Integration product and provides the latest details on what's new in this release.

# Table of Contents

# PREFACE

## 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 .

## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, highlight text in a document and add comments.

### Prerequisites

- You are logged in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, the document is in the **Multi-page HTML** viewing format.

### Procedure

To provide your feedback, perform the following steps:

1. Click the **Feedback** button in the top-right corner of the document to see existing feedback.

   NOTE

   The feedback feature is enabled only in the **Multi-page HTML** format.

2. Highlight the section of the document where you want to provide feedback.
3. Click the **Add Feedback** pop-up that appears near the highlighted text.
   A text box appears in the feedback section on the right side of the page.
4. Enter your feedback in the text box and click **Submit**.
   A documentation issue is created.
5. To view the issue, click the issue link in the feedback view.

# CHAPTER 1. RED HAT INTEGRATION

Red Hat Integration is a comprehensive set of integration and event processing technologies for creating, extending, and deploying container-based integration services across hybrid and multicloud environments. Red Hat Integration provides an agile, distributed, and API-centric solution that organizations can use to connect and share data between applications and systems required in a digital world.

Red Hat Integration includes the following capabilities:

- Real-time messaging
- Cross-datacenter message streaming
- API connectivity
- Application connectors
- Enterprise integration patterns
- API management
- Data transformation
- Service composition and orchestration

**Additional resources**

- Understanding enterprise integration

# CHAPTER 2. CAMEL EXTENSIONS FOR QUARKUS 2.13.3 RELEASE NOTES

## 2.1. CAMEL EXTENSIONS FOR QUARKUS FEATURES

**Fast startup and low RSS memory**

Using the optimized build-time and ahead-of-time (AOT) compilation features of Quarkus, your Camel application can be pre-configured at build time resulting in fast startup times.

**Application generator**

Use the Quarkus application generator to bootstrap your application and discover its extension ecosystem.

**Highly configurable**

All the important aspects of a Camel Extensions for Quarkus application can be set up programmatically with CDI (Contexts and Dependency Injection) or by using configuration properties. By default, a CamelContext is configured and automatically started for you.

Check out the Configuring your Quarkus applications guide for more information on the different ways to bootstrap and configure an application.

**Integrates with existing Quarkus extensions**

Camel Extensions for Quarkus provides extensions for libraries and frameworks that are used by some Camel components which inherit native support and configuration options.

## 2.2. SUPPORTED PLATFORMS, CONFIGURATIONS, DATABASES, AND EXTENSIONS

- For information about supported platforms, configurations, and databases in Camel Extensions for Quarkus version 2.13.2, see the Supported Configuration page on the Customer Portal (login required).

- For a list of Red Hat Camel Extensions for Quarkus extensions and the Red Hat support level for each extension, see the Extensions Overview chapter of the *Camel Extensions for Quarkus Reference* (login required).

## 2.3. BOM FILES FOR CAMEL EXTENSIONS FOR QUARKUS

- To configure your Red Hat Camel Extensions for Quarkus version 2.13.3 projects to use the supported extensions, use the latest Bill Of Materials (BOM) version 2.13.7.SP3-redhat-00003 or newer, from the Redhat Maven Repository.

For more information about BOM dependency management, see Developing Applications with Camel Extensions for Quarkus

## 2.4. TECHNOLOGY PREVIEW EXTENSIONS

Items designated as Technology Preview in the Extensions Overview chapter of the *Camel Extensions for Quarkus Reference* have limited supportability, as defined by the Technology Preview Features Support Scope.

NOTE

Although CXF is fully supported, issues remain with this release of Camel Extensions for Quarkus. The following features are still in Technology Preview:

**CEQ-5390 WS-ReliableMessaging**

Full support for CXF WS-ReliableMessaging is currently unavailable, and it remains in Technology Preview for 2.13.3.

## 2.5. KNOWN ISSUES

**CEQ-6263 OpenTelemetry traces not being generated in sequence**

If you try to capture DB call traces through OpenTelemetry, the bean call executes in a new thread, and a new OpenTelemetry Context is created in the JBDC driver, with a new span outside the expected parent.

NOTE

To ensure that the sequence of recorded spans is correct, you must use the full **to("bean:")** endpoint URI and not the shortened **.bean()** EIP DSL method.

**CEQ-6217 Templated route fails if it is processed before route template**

If the templated route definition is set in **camel.main.routes-include-pattern** before the route template definition, the Camel application will fail to start.

**CEQ-6203 CXF on OpenShift requires quarkus.cxf.path property**

When you deploy Camel Quarkus with CXF on OpenShift, you must set the **quarkus.cxf.path** property.
Otherwise, the pod will not work due to missing liveness/readiness probes:

**Example**

```
11:36:07,343 Can't find the request for http://10.10.10.10:8080/q/health/ready's Observer
```

**CEQ-5705 Extension camel-quarkus-snmp not supported in Native**

In Camel Extensions for Quarkus we support the **camel-quarkus-snmp** component in JVM mode only.
SNMP version 3 is supported only for the operation **poll**. (This limitation is caused by an issue in Camel 3.18.6. For more information, see CAMEL-19298).

NOTE

Currently, http://code.quarkus.redhat.com may not list **camel-quarkus-snmp**, but we do in fact support the component in JVM mode.

## 2.6. IMPORTANT NOTES

**Support for AdoptiumJDK**

Camel Extensions for Quarkus version 2.13.3 includes support for AdoptiumJDK 11 and AdoptiumJDK 17.

### Camel upgraded from version 3.14.2 to version 3.18.6

Camel Extensions for Quarkus version 2.13.3 has been upgraded from Camel version 3.14.2 to Camel version 3.18.6. For additional information about each intervening Camel patch release, refer to the following:

- Apache Camel 3.14.3 Release Notes
- Apache Camel 3.14.4 Release Notes
- Apache Camel 3.14.5 Release Notes
- Apache Camel 3.14.6 Release Notes
- Apache Camel 3.14.7 Release Notes
- Apache Camel 3.15.0 Release Notes
- Apache Camel 3.16.0 Release Notes
- Apache Camel 3.17.0 Release Notes
- Apache Camel 3.18.0 Release Notes
- Apache Camel 3.18.1 Release Notes
- Apache Camel 3.18.2 Release Notes
- Apache Camel 3.18.3 Release Notes
- Apache Camel 3.18.4 Release Notes
- Apache Camel 3.18.5 Release Notes
- Apache Camel 3.18.6 Release Notes

### Camel Quarkus upgraded from version 2.7 to version 2.13

Camel Extensions for Quarkus version 2.13.3 has been upgraded from Camel Quarkus version 2.7 to Camel Quarkus version 2.13. For additional information about each intervening Camel Quarkus patch release, refer to the following:

- Apache Camel Quarkus 2.8.0 Release Notes
- Apache Camel Quarkus 2.9.0 Release Notes
- Apache Camel Quarkus 2.10.0 Release Notes
- Apache Camel Quarkus 2.11.0 Release Notes
- Apache Camel Quarkus 2.12.0 Release Notes
- Apache Camel Quarkus 2.13.0 Release Notes
- Apache Camel Quarkus 2.13.1 Release Notes
- Apache Camel Quarkus 2.13.2 Release Notes
- Apache Camel Quarkus 2.13.3 Release Notes

## 2.7. RESOLVED ISSUES

The following table lists known issues that were affecting Camel Extensions for Quarkus, which have been fixed in Camel Extensions for Quarkus version 2.13.3.

Table 2.1. Resolved issues

| Issue | Description |
|---|---|
| CEQ-6263 | OpenTelemetry traces not being generated in sequence. |
| CEQ-6254 | Support extension: camel-quarkus-vertx-http |
| CEQ-5705 | Support extension: camel-quarkus-snmp |
| CEQ-5265 | Support extension: camel-quarkus-yaml-dsl |
| CEQ-4706 | Support extension: camel-quarkus-amqp |
| CEQ-4618 | Support extension: camel-quarkus-jdbc |
| CEQ-2320 | XML DSL Language - extension camel-quarkus-xml-io-dsl |

### 2.7.1. Previous releases

For details of issues resolved between Camel Quarkus 2.7 and Camel Quarkus 2.13, see the Release Notes for each patch release.

## 2.8. DEPRECATED CAMEL EXTENSIONS FOR QUARKUS FEATURES

The following capabilities and certifications not available in the next major release of CEQ 3.x, and are deprecated in this release.

### 2.8.1. JDK 11

#### Deprecated features

JDK 11 is deprecated in this release of Camel Extensions for Quarkus. It is not supported in future releases.

### 2.8.2. Camel-microprofile-metrics

**Camel-microprofile-metrics** is deprecated since Camel Extensions for Quarkus version 2.13.2. Use the **camel-micrometer** instead.

## 2.9. EXTENSIONS ADDED IN THIS RELEASE

The following table lists the extensions added in this release of Camel Extensions for Quarkus version Camel Extensions for Quarkus .

Table 2.2. Added extensions

| Extension | Artifact | Description | Note |
|---|---|---|---|
| AMQP | **camel-quarkus-amqp** | Messaging with AMQP protocol using Apache QPid Client. | |
| CXF | **camel-quarkus-cxf-soap** | Expose SOAP WebServices using Apache CXF or connect to external WebServices using CXF WS client. | |
| JDBC | **camel-quarkus-jdbc** | Access databases through SQL and JDBC. | |
| SNMP | **camel-quarkus-snmp** | Receive traps and poll SNMP (Simple Network Management Protocol) capable devices. | SNMP version 3 is supported only for the operation **poll**. (This limitation is caused by an issue in Camel 3.18.6. For more information, see CAMEL-19298). |
| Vert.x HTTP Client | **camel-quarkus-vertx-http** | Camel HTTP client support with Vert.x. | |
| YAML -DSL | **camel-quarkus-yaml-dsl** | A YAML stack for parsing YAML route definitions. | |
| XML IO DSL | **camel-quarkus-xml-io-dsl** | An XML stack for parsing XML route definitions | |

## 2.10. DATA FORMATS ADDED IN THIS RELEASE

No data formats have been added in the version Camel Extensions for Quarkus release of 2.13.3.

## 2.11. ADDITIONAL RESOURCES

- Supported Configurations
- Camel Extensions for Quarkus Reference
- Getting Started with Camel Extensions for Quarkus
- Developing Applications with Camel Extensions for Quarkus

# CHAPTER 3. DEBEZIUM 2.1.4 RELEASE NOTES

Debezium is a distributed change data capture platform that captures row-level changes that occur in database tables and then passes corresponding change event records to Apache Kafka topics. Applications can read these *change event streams* and access the change events in the order in which they occurred. Debezium is built on Apache Kafka and is deployed and integrated with AMQ Streams on OpenShift Container Platform or on Red Hat Enterprise Linux.

The following topics provide release details:

- Section 3.1, "Debezium database connectors"
- Section 3.2, "Debezium supported configurations"
- Section 3.3, "Debezium installation options"
- Section 3.4, "Upgrading Debezium from version 1.x to 2.1.4"
- Section 3.5, "New Debezium features"

## 3.1. DEBEZIUM DATABASE CONNECTORS

Debezium provides connectors based on Kafka Connect for the following common databases:

- Db2
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server

### 3.1.1. Connector usage notes

- Db2
  - The Debezium Db2 connector does not include the Db2 JDBC driver (**jcc-11.5.0.0.jar**). See the deployment instructions for information about how to deploy the necessary JDBC driver.
  - The Db2 connector requires the use of the abstract syntax notation (ASN) libraries, which are available as a standard part of Db2 for Linux.
  - To use the ASN libraries, you must have a license for IBM InfoSphere Data Replication (IIDR). You do not have to install IIDR to use the libraries.
- MongoDB
  - Currently, you cannot use the transaction metadata feature of the Debezium MongoDB connector with MongoDB 4.2.
- Oracle

  - The Debezium Oracle connector does not include the Oracle JDBC driver (**ojdbc8.jar**). See the deployment instructions for information about how to deploy the necessary JDBC driver.

- PostgreSQL
  - To use the Debezium PostgreSQL connector you must use the **pgoutput** logical decoding output plug-in, which is the default for PostgreSQL versions 10 and later.

**Additional resources**

- Getting Started with Debezium
- Debezium User Guide

## 3.2. DEBEZIUM SUPPORTED CONFIGURATIONS

For information about Debezium supported configurations, including information about supported database versions, see the Debezium 2.1.4 Supported configurations page .

### 3.2.1. AMQ Streams API version

Debezium runs on AMQ Streams 2.3.

AMQ Streams supports the **v1beta2** API version, which updates the schemas of the AMQ Streams custom resources. Older API versions are deprecated. After you upgrade to AMQ Streams 1.7, but before you upgrade to AMQ Streams 1.8 or later, you must upgrade your custom resources to use API version **v1beta2**.

For more information, see the Debezium User Guide.

## 3.3. DEBEZIUM INSTALLATION OPTIONS

You can install Debezium with AMQ Streams on OpenShift or on Red Hat Enterprise Linux:

- Installing Debezium on OpenShift
- Installing Debezium on RHEL

## 3.4. UPGRADING DEBEZIUM FROM VERSION 1.X TO 2.1.4

The current version of Debezium includes changes that require you to follow specific steps when you upgrade from an earlier version. For more information, refer to the list of breaking changes and the upgrade procedure.

### 3.4.1. Upgrading connectors to Debezium 2.1.4

Debezium 2.1.4 is the first Red Hat release of a new Debezium major release version. Some of the changes in the Debezium 2.1.4 are not backward-compatible with previous versions of Debezium. As a result, to preserve data and ensure continued operation when you upgrade from Debezium 1.x versions to 2.1.4, you must complete some manual steps during the upgrade process.

One significant change is that the names of some connector parameters have changed. To accommodate these changes, review the configuration properties updates , and note the properties that are present in your connector configuration. Before you upgrade, edit the configuration of each Debezium connector to add the new names of any changed properties. Before you upgrade, edit the configuration of any 1.x connector instances so that both the old and new property names are present. After the upgrade, you can remove the old configuration options.

### Prerequisites

- Debezium is now compatible with Kafka versions up to 3.3.1. This is the default Kafka version in AMQ Streams 2.3.
- The Java 11 runtime is required and must be available prior to upgrading. AMQ Streams 2.3 supports Java 11. Use Java 11 when developing new applications. Java 11 enables use of recent language updates, such as the new String API and changes in predicate support, while also benefiting from Java performance improvements. Java 8 is no longer supported in AMQ Streams 2.3.
- Check the backward-incompatible changes in the Breaking changes list.
- Verify that your environment complies with the Debezium 2.1.4 Supported Configurations.

### Procedure

1. From the OpenShift console, review the Kafka Connector YAML to identify the connector configuration that are no longer valid in Debezium 2.1.4. Refer to Table 3.1, “Updates to connector configuration properties” for details.
2. Edit the configuration to add the 2.x equivalents for the properties that you identify in Step 1, so that both the old and new property names are present. Set the values of the new properties to the values that were previously specified for the old properties.
3. From the OpenShift console, stop Kafka Connect to gracefully stop the connector.
4. From the OpenShift console, edit the Kafka Connect image YAML to reference the Debezium 2.1.4.Final version of the connector zip file.
5. From the OpenShift console, edit the Kafka Connector YAML to remove any configuration options that are no longer valid for your connector.
6. Adjust your application’s storage dependencies, as needed, depending on the storage module implementation dependencies in your code. See Changes to Debezium storage in the list of Breaking changes.
7. Restart Kafka Connect to start the connector. After you restart the connector, the 2.1.4.Final connector continues to process events from the point at which you stopped the connector before the upgrade. Change events records that the connector wrote to Kafka before the upgrade are not modified.

## 3.5. NEW DEBEZIUM FEATURES

Debezium 2.1.4 includes the following updates.

- Section 3.5.1, “Breaking changes”
- Section 3.5.2, “Features promoted to General Availability”

### 3.5.1. Breaking changes

The following changes in Debezium 2.1.4 represent significant differences in connector behavior and require configuration changes that are not compatible with earlier Debezium versions:

**Changes that apply to multiple connectors**

**Database history topic**

Now referred to as the *database schema history* topic.

**Limits on object sizes for memory queues**

Sizes are no longer calculated by using reflection. Instead, queue limits are estimated based on the message schema. (DBZ-2766) (MongodB, MySQL, Oracle, PostgreSQL, SQL Server )

**Exposure of connector metrics**

Debezium previously exposed connector metrics as a single tuple of snapshot, streaming, and history-based beans. With this release, connector metrics are now exposed as a multi-partition scheme. As a result, metrics names, and the way in which they are exposed is changed (DBZ-4726). If you use Grafana, Prometheus, or similar JMX frameworks for gathering metrics, review your process for collecting metrics.

**database.server.name property**

No longer used in the connector configuration. For more information, see Table 3.1, “Updates to connector configuration properties”.

**Schema definition**

For naming and versioning consistency, Debezium schemas are now defined in a central point (DBZ-4365, DBZ-5044). If you use a schema registry, schema compatibility issues might occur.

**Debezium storage changes**

In previous releases, Integration supported reading and storing offsets, history, and other data as a part of the debezium-core module. This release includes a new **debezium-storage** module with implementations for storing data in a local file system or in Kafka (DBZ-5229). The extension point implemented in this approach makes it possible to introduce other storage implementations in the future. As part of the upgrade, you might need to adjust your application’s dependencies depending on the storage module implementations required by the code.

**Restart after communication exceptions**

After an exception related to communication (SqlException, IOException) is thrown, by default, the Debezium MongoDB, MySQL, PostgreSQL, and SQL Server connectors now restart automatically (DBZ-5244).

**Default value of theskipped.operations configuration option**

The default value is now **truncate** (DBZ-5497) (MongoDB, MySQL, Oracle, PostgreSQL, SQL Server)

**Default value ofschema.name.adjustment.mode property**

The default value is now **none** (DBZ-5541). The previous default option,  **avro** was a good choice for customers who use the Avro converter, but it caused confusion in environments that use the JSON converter. As part of this change, the **sanitize.field.names** property is no longer available.

*** Removal of connector configuration properties**

Several properties that were available in Debezium 1.x versions are no longer valid and have been replaced by new properties. For more information, see the following table:

**Table 3.1. Updates to connector configuration properties**

| 1.x property | Equivalent 2.x property |
|---|---|
| **database.*** (pass-through database driver properties) (DBZ-5043) | **driver.*** |
| **database.dbname** (SQL Server) | **database.names** |
| **database.history.consumer.*** (DBZ-5043) | **schema.history.internal.consumer.*** |
| **database.history.kafka.bootstrap.servers** (DBZ-5043) | **schema.history.internal.kafka.bootstrap.servers** |
| **database.history.kafka.topic** (DBZ-5043) | **schema.history.internal.kafka.topic** |
| **database.history.producer.*** (DBZ-5043) | **schema.history.internal.producer.*** |
| **database.server.name** (DBZ-5043) | **topic.prefix** |
| **mongodb.name** (MongoDB) | **topic.prefix** |
| **schema_blacklist** (DBZ-5045) | **schema_exclude_list** |
| **schema_whitelist** (DBZ-5045) | **schema_include_list** |

### Changes that apply to the MySQL connector

- The MySQL connector no longer supports legacy JDBC legacy date/time properties (DBZ-4965).

### Changes that apply to the MongoDB connector

- The MongoDB connector no longer supports streaming directly from the oplog. Change streams represents a superior mechanism for performing change data capture with MongoDB. Rather than reading the oplog directly, the connector now delegates the task of capturing and decoding the oplog data to MongoDB change streams, which expose the changes that occur within a collection as an event stream. The Debezium connector subscribes to the stream and then delivers the changes downstream to Kafka. The transition to change streams offers a variety of benefits, including the ability to stream changes from non-primary nodes, and the ability to emit update events with a full document representation for downstream consumers.
- The configuration property **mongodb.name** is replaced by the **topic.prefix** property.

### Changes that apply to the PostgreSQL connector

- Protocol buffer (**protobuf**) decoding is no longer supported (DBZ-703).
- The **wal2json** plugin is no longer supported (DBZ-4156).

- The PostgreSQL transaction id is now 32-bit integer that rolls over. To simplify de-duplication of transactions, the LSN is now included as part of the identifier (DBZ-5329).

**Changes that apply to the SQL Server connector**

- If SSL is not enabled for a SQL Server database, or if you want to connect to the database without using SSL, disable SSL by setting the value of the **database.encrypt** property in the connector configuration to **false**.
- The **database.dbname** property is replaced by the **database.names** property.

### 3.5.2. Features promoted to General Availability

The following features are promoted from Technology Preview to General Availability in the Debezium 2.1.4 release:

- MongoDB **ExtractNewDocumentState** SMT

### 3.5.3. Debezium feature updates

This Debezium 2.1.4 release provides several feature updates and fixes, including the items in the following list:

- The MySQL connector now supports binlog compression. DBZ-2663
- Limit log output for "Streaming requested from LSN" warnings. DBZ-3007
- Implements support for JSON_TABLE in the MySQL connector parser. DBZ-3575
- You can now pause or stop incremental snapshots by sending a signal. DBZ-4251
- The SQL Server connector now fails when the user account lacks the required CDCReader permission. DBZ-4346
- The MongoDB connector can now decode binary payloads DBZ-4600
- You can now pause and resume a running incremental snapshot DBZ-4727
- You can now specify the MongoDB connection settings by specifying a connection string URI DBZ-4733
- The **field.exclude.list** property for the MongoDB connector now works with fields from different collections that have the same name. DBZ-4846
- The PostgresSQL connector now retries the connection after the error **PSQLException: This connection has been closed.** DBZ-4948
- The MySQL connector now stores the event header timestamp in the the history record DBZ-4998
- The LogMiner batch size is now adjusted based on the current batch size, rather than the default size. DBZ-5005
- You can now configure the maximum number of entries to cache for the ByLogicalTableRouter SMT. DBZ-5072

- A new extension API permits you to query the Debezium version. DBZ-5092
- Adds the field **ts_ms** to schema change events to identify when an event occurs or is processed. DBZ-5098
- When the MongoDB connector converts oplog entries, it now uses the **RawBsonDocument** class rather than **Document**. DBZ-5113
- MySQL commit timestamp DBZ-5170
- The event SCN is now included in Oracle event records. DBZ-5225
- To avoid occurrences of **UnknownTopicOrPartitionException**, you can now set **database.history.kafka.create.timeout.ms** to specify how long the connector waits for the Kafka history topic to be created. DBZ-5249
- After modifying the primary key, LOB type data is now consistent between the source and sink. DBZ-5295
- The MySQL connector now retries after it receives an error when attempting to read the binlog. DBZ-5333
- During an incremental snapshot, the Oracle connector now correctly parses events from a database with a name that includes a period. DBZ-5336
- Support PostgreSQL default value function calls with schema prefixes. DBZ-5340
- The MySQL connector fails to convert unsigned **tinyint** data types for MySQL 8.x. DBZ-5343
- The Oracle connector logs a warning when it detects an unsupported LogMiner operation for a captured table. DBZ-5351
- The Oracle connector throws a NullPointerException when a unique index is based on both system and non-system generated columns. DBZ-5356
- Fixes a problem in which column hash v2 did not work with the MySQL connector. DBZ-5366
- Fixes a problem in which JSON expansion failed for outbox event payloads that contain nested arrays in which the first array contain no elements. DBZ-5367
- Fixes MongoDB connector connection failures when using AWS DocumentDB with MongoDB compatibility. DBZ-5371
- Fixes a problem in which the Oracle connector logged CommitScn in an unexpected format. DBZ-5381
- Fixes the PostgreSQL connector error org.postgresql.util.PSQLException: Bad value for type timestamp/date/time: CURRENT_TIMESTAMP. DBZ-5384
- Fixes a problem with the MySQL connector in which the **previousID** property is missing in the history topic. DBZ-5386
- Check constraint introduces a column based on constraint in the schema change event. DBZ-5390
- Fixes a problem that occurs when the PostgreSQL connector captures a column that is referenced as the PRIMARY KEY, but no matching column is defined in table. DBZ-5398

- Clarifies documentation for **signal.data.collection** when using Oracle with pluggable database support DBZ-5399
- The PostgreSQL connector now uses GMT to specify timestamps. DBZ-5403
- Ad hoc and incremental snapshots now support an **additional-condition** parameter for specifying subsets of data to capture. DBZ-5327
- Adds logic to enable the Oracle connector to gracefully unsupported non-relational tables during streaming DBZ-5441
- The SQL Server connector task now restarts after a "Socket closed" exception. DBZ-5478
- Augment a uniqueness key field/value in regular expression topic naming strategy. DBZ-5480
- MySqlErrorHandler should handle SocketException DBZ-5486
- The MySQL connector now adds database column comments to the connector schema. DBZ-5489
- Expose default values and enum values in schema history messages. DBZ-5511
- Support BASE64_URL_SAFE in BinaryHandlingMode DBZ-5544
- Supply partition when committing offsets with source database DBZ-5557
- Traditional snapshot process setting **source.ts_ms**. DBZ-5591
- Clean up "logical name" configuration. DBZ-5594
- The MySQL Connector now captures TRUNCATE events. DBZ-5610
- Clarify semantics of include/exclude options in documentation. DBZ-5625
- You can now configure the MongoDB connector to include a **before** field when it emits change events. DBZ-5628
- Logging enhancement for non-incremental snapshot in PostgreSQL connector. DBZ-5639
- Improve LogMiner query performance by reducing REGEXP_LIKE disjunctions. DBZ-5648
- You can configure how often the MongoDB connector attempts to send heartbeat messages to the server. DBZ-5736
- Enhance the ability to sanitize topic name DBZ-5790
- You can now configure **flush.lsn.source** to prevent the PostgreSQL connector from automatically committing the LSN of processed records to the database. DBZ-5811
- You can now use the **ComputePartition** SMT to route data to specific topic partitions based on certain fields. DBZ-5847
- You can now configure the **event.processing.failure.handling.mode** to enable the PostgreSQL connector to skip failed LSN checks. DBZ-6012
- Connector offsets now advance correctly in when you use the Oracle connector in pluggable database deployments (CDB) that have infrequent changes. DBZ-6125

## 3.6. TECHNOLOGY PREVIEW FEATURES

IMPORTANT

Technology Preview features are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend implementing any Technology Preview features in production environments. Technology Preview features provide 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.

Debezium includes the following Technology Preview features:

**Parallel initial snapshots**

You can optionally configure SQL-based connectors to use multiple threads when performing an initial snapshot by setting the **snapshot.max.threads** property to a value greater than 1.

**Ad hoc and incremental snapshots for MongoDB connector**

Provides a mechanism for re-running a snapshot of a table for which you previously captured a snapshot.

**CloudEvents converter**

Emits change event records that conform to the CloudEvents specification. The CloudEvents change event envelope can be JSON or Avro and each envelope type supports JSON or Avro as the **data** format. The CloudEvents change event envelope supports Avro encoding change event envelope can be JSON or Avro and each envelope type supports JSON or Avro as the **data** format.

**Content-based routing**

Provides a mechanism for rerouting selected events to specific topics, based on the event content.

**Custom-developed converters**

In cases where the default data type conversions do not meet your needs, you can create custom converters to use with a connector.

**Filter SMT**

Enables you to specify a subset of records that you want the connector to send to the broker.

**Signaling for the MongoDB connector**

Provides a mechanism for modifying the behavior of a connector, or triggering a one-time action, such as initiating an ad hoc snapshot of a table.

**Use of the BLOB, CLOB, and NCLOB data types with the Oracle connector**

The Oracle connector can consume Oracle large object types.

# CHAPTER 4. CAMEL K RELEASE NOTES

Camel K is a lightweight integration framework built from Apache Camel K that runs natively in the cloud on OpenShift. Camel K is specifically designed for serverless and microservice architectures. You can use Camel K to instantly run integration code written in Camel Domain Specific Language (DSL) directly on OpenShift.

Using Camel K with OpenShift Serverless and Knative, containers are automatically created only as needed and are autoscaled under load up and down to zero. This removes the overhead of server provisioning and maintenance and enables you to focus instead on application development.

Using Camel K with OpenShift Serverless and Knative Eventing, you can manage how components in your system communicate in an event-driven architecture for serverless applications. This provides flexibility and creates efficiencies using a publish/subscribe or event-streaming model with decoupled relationships between event producers and consumers.

## 4.1. CAMEL K FEATURES

The Camel K provides cloud-native integration with the following main features:

- Knative Serving for autoscaling and scale-to-zero
- Knative Eventing for event-driven architectures
- Performance optimizations using Quarkus runtime by default
- Camel integrations written in Java or YAML DSL
- Monitoring of integrations using Prometheus in OpenShift
- Quickstart tutorials
- Kamelet Catalog for connectors to external systems such as AWS, Jira, and Salesforce
- Support for Timer and Log Kamelets
- Support for IBM MQ connector
- Support for Oracle 19 database

## 4.2. SUPPORTED CONFIGURATIONS

For information about Camel K supported configurations, standards, and components, see the following Customer Portal articles:

- Camel K Supported Configurations
- Camel K Component Details

### 4.2.1. Camel K Operator metadata

The Camel K includes updated Operator metadata used to install Camel K from the OpenShift OperatorHub. This Operator metadata includes the Operator bundle format for release packaging, which is designed for use with OpenShift Container Platform 4.6 or later.

### Additional resources

- Operator bundle format in the OpenShift documentation .

## 4.3. IMPORTANT NOTES

Important notes for the Red Hat Integration - Camel K release:

**Removing support of metering labels from Red Hat Integration - Camel K**

Metering labels for Camel K Operator and pods are no longer supported.

**Security update for Red Hat Integration - Camel K**

For details on how to apply this update, see How do I apply package updates to my RHEL system?

> NOTE
>
> You must apply all the previously release Errata upgrades to your system before applying this security update.

**Support to run Camel K on ROSA**

Camel K is now supported to run on Red Hat OpenShift Service on AWS (ROSA).

**Support for IBM MQ source connector in Camel K**

IBM MQ source connector kamelet is added to latest Camel K.

**Support for Oracle 19**

Oracle 19 is now supported in Camel K. Refer Supported configurations page for more information.

**Using Camel K CLI commands on Windows machine**

When using kamel cli commands on Windows machine, the path in the **resource** option in the command must use linux format. For example:

```
//Windows path
kamel run file.groovy --dev --resource file:C:\user\folder\tempfile@/tmp/file.txt

//Must be converted to
kamel run file.groovy --dev --resource file:C:/user/folder/tempfile@/tmp/file.txt
```

**Red Hat Integration - Camel K Operator image size is increased**

Since Red Hat Integration - Camel K 1.10.1.redhat-00026, the size of the Camel K Operator image is doubled.

**Accepted Camel case notations in YAML DSL**

Since Red Hat Integration - Camel K 1.10.1.redhat-00026, the YAML DSL will accept camel case notation (i.e, **setBody**) as well as snake case (i.e **set-body**). Please note that there are some differences in the syntax as schema is subject to changes within Camel versions.

## 4.4. SUPPORTED CAMEL QUARKUS EXTENSIONS

This section lists the Camel Quarkus extensions that are supported for this release of Camel K (only when used inside a Camel K application).

NOTE

These Camel Quarkus extensions are supported only when used inside a Camel K application. These Camel Quarkus extensions are not supported for use in standalone mode (without Camel K).

### 4.4.1. Supported Camel Quarkus connector extensions

The following table shows the Camel Quarkus connector extensions that are supported for this release of Camel K (only when used inside a Camel K application).

| Name | Package |
|---|---|
| Attachments | **camel-quarkus-attachments** |
| AWS DynamoDB | **camel-quarkus-aws-ddb** |
| AWS 2 Kinesis | **camel-quarkus-aws2-kinesis** |
| AWS 2 Lambda | **camel-quarkus-aws2-lambda** |
| AWS 2 S3 Storage Service | **camel-quarkus-aws2-s3** |
| AWS 2 Simple Notification System (SNS) | **camel-quarkus-aws2-sns** |
| AWS 2 Simple Queue Service (SQS) | **camel-quarkus-aws2-sqs** |
| Azure Storage Blob (Technology Preview) | **camel-quarkus-azure-storage-blob** |
| Azure Storage Queue (Technology Preview) | **camel-quarkus-azure-storage-queue** |
| Bean | **camel-quarkus-bean** |
| Bean Validator | **camel-quarkus-bean-validator** |
| Browse | **camel-quarkus-browse** |
| Cassandra CQL | **camel-quarkus-cassandraql** |
| Core | **camel-quarkus-core** |
| Cron | **camel-quarkus-cron** |
| CXF | **camel-quarkus-cxf-soap** |
| Dataformat | **camel-quarkus-dataformat** |
| Direct | **camel-quarkus-direct** |

| Name | Package |
|---|---|
| File | **camel-quarkus-file** |
| FHIR | **camel-quarkus-fhir** |
| FTP | **camel-quarkus-ftp** |
| Google BigQuery | **camel-quarkus-google-bigquery** |
| Google Pubsub | **camel-quarkus-google-pubsub** |
| HTTP | **camel-quarkus-http** |
| Infinispan | **camel-quarkus-infinispan** |
| Jira | **camel-quarkus-jira** |
| JMS | **camel-quarkus-jms** |
| JPA | **camel-quarkus-jpa** |
| JTA | **camel-quarkus-jta** |
| Kafka | **camel-quarkus-kafka** |
| Kamelet | **camel-quarkus-kamelet** |
| Kubernetes | **camel-quarkus-kubernetes** |
| Log | **camel-quarkus-log** |
| Mail | **camel-quarkus-mail** |
| MicroProfile Fault Tolerance | **camel-quarkus-microprofile-fault-tolerance** |
| MicroProfile Health | **camel-quarkus-microprofile-health** |
| MicroProfile Metrics | **camel-quarkus-microprofile-metrics** |
| MLLP | **camel-quarkus-mllp** |
| Mock | **camel-quarkus-mock** |
| MongoDB | **camel-quarkus-mongodb** |
| Netty | **camel-quarkus-netty** |

| Name | Package |
|---|---|
| OpenAPI Java | **camel-quarkus-openapi-java** |
| Paho | **camel-quarkus-camel-quarkus-paho** |
| Paho MQTT 5 | **camel-quarkus-paho-mqtt5** |
| Platform HTTP | **camel-quarkus-platform-http** |
| Quartz | **camel-quarkus-quartz** |
| Rest | **camel-quarkus-rest** |
| REST OpenApi | **camel-quarkus-rest-openapi** |
| Salesforce | **camel-quarkus-salesforce** |
| SEDA | **camel-quarkus-seda** |
| Slack | **camel-quarkus-slack** |
| SQL | **camel-quarkus-sql** |
| Telegram | **camel-quarkus-telegram** |
| Timer | **camel-quarkus-timer** |
| Validator | **camel-quarkus-validator** |
| XQuery | **camel-quarkus-saxon** |
| Zip File | **camel-quarkus-zipfile** |

### 4.4.2. Supported Camel Quarkus dataformat extensions

The following table shows the Camel Quarkus dataformat extensions that are supported for this release of Camel K (only when used inside a Camel K application).

| Name | Package |
|---|---|
| Avro | **camel-quarkus-avro** |
| Avro Jackson | **camel-quarkus-jackson-avro** |
| Bindy (for CSV) | **camel-qaurkus-bindy** |

| Name | Package |
|---|---|
| HL7 | **camel-quarkus-hl7** |
| Jackson | **camel-quarkus-jackson** |
| JacksonXML | **camel-quarkus-jacksonxml** |
| JAXB | **camel-quarkus-jaxb** |
| JSON Gson | **camel-quarkus-gson** |
| Protobuf Jackson | **camel-quarkus-jackson-protobuf** |
| SOAP dataformat | **camel-quarkus-soap** |

### 4.4.3. Supported Camel Quarkus language extensions

In this release, Camel K supports the following Camel Quarkus language extensions (for use in Camel expressions and predicates):

- Bean method
- Constant
- ExchangeProperty
- File
- Header
- HL7 Terser
- Ref
- Simple
- Tokenize
- JsonPath
- XPath
- XQuery

### 4.4.4. Supported Camel K traits

In this release, Camel K supports the following Camel K traits.

- Builder trait
- Camel trait

- Container trait
- Dependencies trait
- Deployer trait
- Deployment trait
- Environment trait
- Jvm trait
- Kamelets trait
- Owner trait
- Platform trait
- Pull Secret trait
- Prometheus trait
- Quarkus trait
- Route trait
- Service trait
- Error Handler trait

## 4.5. SUPPORTED KAMELETS

The following table lists the kamelets that are provided as OpenShift resources when you install the Camel K operator.

For details about these kamelets, go to: https://github.com/openshift-integration/kamelet-catalog/tree/kamelet-catalog-1.8

For information about how to use kamelets to connect applications and services, see https://access.redhat.com/documentation/en-us/red_hat_integration/2022.q3/html-single/integrating_applications_with_kamelets.

### IMPORTANT

Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production.

These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview.

**Table 4.1. Kamelets provided with the Camel K operator**

| Kamelet | File name | Type (Sink, Source, Action) |
|---|---|---|
| Ceph sink | **ceph-sink.kamelet.yaml** | Sink |
| Ceph Source | **ceph-source.kamelet.yaml** | Source |
| Jira Add Comment sink | **jira-add-comment-sink.kamelet.yaml** | Sink |
| Jira Add Issue sink | **jira-add-issue-sink.kamelet.yaml** | Sink |
| Jira Transition Issue sink | **jira-transition-issue-sink.kamelet.yaml** | Sink |
| Jira Update Issue sink | **jira-update-issue-sink.kamelet.yaml** | Sink |
| Avro Deserialize action | **avro-deserialize-action.kamelet.yaml** | Action (data conversion) |
| Avro Serialize action | **avro-serialize-action.kamelet.yaml** | Action (data conversion) |
| AWS DynamoDB sink | **aws-ddb-sink.kamelet.yaml** | Sink |
| AWS Redshift sink | **aws-redshift-sink.kamelet.yaml** | Sink |
| AWS 2 Kinesis sink | **aws-kinesis-sink.kamelet.yaml** | Sink |
| AWS 2 Kinesis source | **aws-kinesis-source.kamelet.yaml** | Source |
| AWS 2 Lambda sink | **aws-lambda-sink.kamelet.yaml** | Sink |
| AWS 2 Simple Notification System sink | **aws-sns-sink.kamelet.yaml** | Sink |
| AWS 2 Simple Queue Service sink | **aws-sqs-sink.kamelet.yaml** | Sink |
| AWS 2 Simple Queue Service source | **aws-sqs-source.kamelet.yaml** | Source |
| AWS 2 Simple Queue Service FIFO sink | **aws-sqs-fifo-sink.kamelet.yaml** | Sink |
| AWS 2 S3 sink | **aws-s3-sink.kamelet.yaml** | Sink |
| AWS 2 S3 source | **aws-s3-source.kamelet.yaml** | Source |
| AWS 2 S3 Streaming Upload sink | **aws-s3-streaming-upload-sink.kamelet.yaml** | Sink |

| Kamelet | File name | Type (Sink, Source, Action) |
|---|---|---|
| Azure Storage Blob Source (Technology Preview) | **azure-storage-blob-source.kamelet.yaml** | Source |
| Azure Storage Blob Sink (Technology Preview) | **azure-storage-blob-sink.kamelet.yaml** | Sink |
| Azure Storage Queue Source (Technology Preview) | **azure-storage-queue-source.kamelet.yaml** | Source |
| Azure Storage Queue Sink (Technology Preview) | **azure-storage-queue-sink.kamelet.yaml** | Sink |
| Cassandra sink | **cassandra-sink.kamelet.yaml** | Sink |
| Cassandra source | **cassandra-source.kamelet.yaml** | Source |
| Extract Field action | **extract-field-action.kamelet.yaml** | Action |
| FTP sink | **ftp-sink.kamelet.yaml** | Sink |
| FTP source | **ftp-source.kamelet.yaml** | Source |
| Has Header Key Filter action | **has-header-filter-action.kamelet.yaml** | Action (data transformation ) |
| Hoist Field action | **hoist-field-action.kamelet.yaml** | Action |
| HTTP sink | **http-sink.kamelet.yaml** | Sink |
| Insert Field action | **insert-field-action.kamelet.yaml** | Action (data transformation ) |
| Insert Header action | **insert-header-action.kamelet.yaml** | Action (data transformation ) |
| Is Tombstone Filter action | **is-tombstone-filter-action.kamelet.yaml** | Action (data transformation ) |
| Jira source | **jira-source.kamelet.yaml** | Source |
| JMS sink | **jms-amqp-10-sink.kamelet.yaml** | Sink |

| Kamelet | File name | Type (Sink, Source, Action) |
|---|---|---|
| JMS source | **jms-amqp-10-source.kamelet.yaml** | Source |
| JMS IBM MQ sink | **jms-ibm-mq-sink.kamelet.yaml** | Sink |
| JMS IBM MQ source | **jms-ibm-mq-source.kamelet.yaml** | Source |
| JSON Deserialize action | **json-deserialize-action.kamelet.yaml** | Action (data conversion) |
| JSON Serialize action | **json-serialize-action.kamelet.yaml** | Action (data conversion) |
| Kafka sink | **kafka-sink.kamelet.yaml** | Sink |
| Kafka source | **kafka-source.kamelet.yaml** | Source |
| Kafka Topic Name Filter action | **topic-name-matches-filter-action.kamelet.yaml** | Action (data transformation ) |
| Log sink (for development and testing purposes) | **log-sink.kamelet.yaml** | Sink |
| MariaDB sink | **mariadb-sink.kamelet.yaml** | Sink |
| Mask Fields action | **mask-field-action.kamelet.yaml** | Action (data transformation ) |
| Message TimeStamp Router action | **message-timestamp-router-action.kamelet.yaml** | Action (router) |
| MongoDB sink | **mongodb-sink.kamelet.yaml** | Sink |
| MongoDB source | **mongodb-source.kamelet.yaml** | Source |
| MySQL sink | **mysql-sink.kamelet.yaml** | Sink |
| PostgreSQL sink | **postgresql-sink.kamelet.yaml** | Sink |
| Predicate filter action | **predicate-filter-action.kamelet.yaml** | Action (router/filter) |

| Kamelet | File name | Type (Sink, Source, Action) |
|---|---|---|
| Protobuf Deserialize action | **protobuf-deserialize-action.kamelet.yaml** | Action (data conversion) |
| Protobuf Serialize action | **protobuf-serialize-action.kamelet.yaml** | Action (data conversion) |
| Regex Router action | **regex-router-action.kamelet.yaml** | Action (router) |
| Replace Field action | **replace-field-action.kamelet.yaml** | Action |
| Salesforce Create | **salesforce-create-sink.kamelet.yaml** | Sink |
| Salesforce Delete | **salesforce-delete-sink.kamelet.yaml** | Sink |
| Salesforce Update | **salesforce-update-sink.kamelet.yaml** | Sink |
| SFTP sink | **sftp-sink.kamelet.yaml** | Sink |
| SFTP source | **sftp-source.kamelet.yaml** | Source |
| Slack source | **slack-source.kamelet.yaml** | Source |
| SQL Server Database sink | **sqlserver-sink.kamelet.yaml** | Sink |
| Telegram source | **telegram-source.kamelet.yaml** | Source |
| Throttle action | **throttle-action.kamelet.yaml** | Action |
| Timer source (for development and testing purposes) | **timer-source.kamelet.yaml** | Source |
| TimeStamp Router action | **timestamp-router-action.kamelet.yaml** | Action (router) |
| Value to Key action | **value-to-key-action.kamelet.yaml** | Action (data transformation ) |

## 4.6. CAMEL K KNOWN ISSUES

The following known issues apply to the Camel K:

**ENTESB-15306 - CRD conflicts between Camel K and Fuse Online**

If an older version of Camel K has ever been installed in the same OpenShift cluster, installing Camel K from the OperatorHub fails due to conflicts with custom resource definitions. For example, this includes older versions of Camel K previously available in Fuse Online.

For a workaround, you can install Camel K in a different OpenShift cluster, or enter the following command before installing Camel K:

```
$ oc get crds -l app=camel-k -o json | oc delete -f -
```

### ENTESB-15858 - Added ability to package and run Camel integrations locally or as container images

Packaging and running Camel integrations locally or as container images is not currently included in the Camel K and has community-only support.

For more details, see the Apache Camel K community .

### ENTESB-16477 - Unable to download jira client dependency with productized build

When using Camel K operator, the integration is unable to find dependencies for jira client. The work around is to add the atlassian repo manually.

```
apiVersion: camel.apache.org/v1
kind: IntegrationPlatform
metadata:
  labels:
    app: camel-k
  name: camel-k
spec:
  configuration:
  - type: repository
    value: <atlassian repo here>
```

### ENTESB-17033 - Camel-K ElasticsearchComponent options ignored

When configuring the Elasticsearch component, the Camel K ElasticsearchComponent options are ignored. The work around is to add **getContext().setAutowiredEnabled(false)** when using the Elasticsearch component.

### ENTESB-17061 - Can’t run mongo-db-source kamelet route with non-admin user - Failed to start route mongodb-source-1 because of null

It is not possible to run **mongo-db-source kamelet** route with non-admin user credentials. Some part of the component require admin credentials hence it is not possible run the route as a non-admin user.

## 4.7. CAMEL K FIXED ISSUES

The following sections list the issues that have been fixed in Red Hat Integration - Camel K 1.10.1.redhat-00026:

- Section 4.7.1, “Feature requests in Camel K 1.10.1.redhat-00026”
- Section 4.7.2, “Bugs resolved in Camel K 1.10.1.redhat-00026”

### 4.7.1. Feature requests in Camel K 1.10.1.redhat-00026

The following table lists the feature requests in Camel K 1.10.1.redhat-00026:

**Table 4.2. Camel K 1.10.1.redhat-00026 feature requests**

| Issue | Description |
|---|---|
| CMLK-590 | Trimming extracted String data capability by "extract-field-action" |

### 4.7.2. Bugs resolved in Camel K 1.10.1.redhat-00026

The following table lists the resolved bugs in Camel K 1.10.1.redhat-00026:

**Table 4.3. Camel K 1.10.1.redhat-00026 Resolved Bugs**

| Issue | Description |
|---|---|
| CMLK-774 | "runtime error: invalid memory address or nil pointer dereference" from kamel promote command |
| CMLK-653 | Kamel uninstall does not remove IntegrationPlatform in case of global operator |
| CMLK-554 | Regression - CK and Knative fighting over the KService |
| CMLK-277 | Dockerfile fuse-camel-k-prod-operator-metadata contains label v4.6 |
| CMLK-275 | Mismatch in supported Kamelets vs Camel components |
| CMLK-560 | Invalid Semantic Version during the 'kamel promote' command |
| CMLK-268 | Jira source - duplicate messages when more than one issue is created within the poll delay |
| CMLK-161 | Kamelet-catalog: Some of the version placeholder are not managed in the BOM |
| CMLK-610 | CVE-2023-1436 jettison: Uncontrolled Recursion in JSONArray [rhint-camel-k-1] |
| CMLK-721 | CVE-2021-46877 jackson-databind: Possible DoS if using JDK serialization to serialize JsonNode [rhint-camel-k-1] |
| CMLK-120 | CVE-2022-4492 undertow: Server identity in https connection is not checked by the undertow client [rhint-camel-k-1] |
| CMLK-118 | CVE-2022-24999 qs: express: "qs" prototype poisoning causes the hang of the node process [rhint-camel-k-1] |
| CMLK-93 | ClassNotFoundException Jira kamelet binding in native mode |

| Issue | Description |
|---|---|
| CMLK-141 | CVE-2022-41946 jdbc-postgresql: postgresql-jdbc: Information leak of prepared statement data due to insecure temporary file permissions [rhint-camel-k-1] |
| CMLK-117 | CVE-2022-46363 CXF: Apache CXF: directory listing / code exfiltration [rhint-camel-k-1] |
| CMLK-133 | CVE-2022-1471 snakeyaml: Constructor Deserialization Remote Code Execution [rhint-camel-k-1] |
| CMLK-142 | CVE-2022-4245 codehaus-plexus: XML External Entity (XXE) Injection [rhint-camel-k-1] |
| CMLK-137 | CVE-2022-4244 codehaus-plexus: Directory Traversal [rhint-camel-k-1] |
| CMLK-124 | CVE-2022-39368 scandium: Failing DTLS handshakes may cause throttling to block processing of records [rhint-camel-k-1] |
| CMLK-628 | CVE-2023-1370 json-smart: Uncontrolled Resource Consumption vulnerability in json-smart (Resource Exhaustion) [rhint-camel-k-1.10] |
| CMLK-815 | CVE-2022-42003 Vulnerable artifact jackson-databind is present in CK 1.10.1 CK6 MRRC repo zip file |

# CHAPTER 5. CAMEL SPRING BOOT RELEASE NOTES

## 5.1. CAMEL SPRING BOOT FEATURES

Camel Spring Boot introduces Camel support for Spring Boot which provides auto-configuration of the Camel and starters for many Camel components. The opinionated auto-configuration of the Camel context auto-detects Camel routes available in the Spring context and registers the key Camel utilities (like producer template, consumer template and the type converter) as beans.

## 5.2. SUPPORTED PLATFORMS, CONFIGURATIONS, DATABASES, AND EXTENSIONS FOR CAMEL SPRING BOOT

- For information about supported platforms, configurations, and databases in Camel Spring Boot, see the Supported Configuration page on the Customer Portal (login required).

- For a list of Red Hat Camel Spring Boot extensions, see the *Camel Spring Boot Reference* (login required).

## 5.3. IMPORTANT NOTES

Documentation for Camel Spring Boot components is available in the Camel Spring Boot Reference. Documentation for additional Camel Spring Boot components will be added to this reference guide.

**Migration from Fuse 7.11 to Camel Spring Boot**

This release contains a Migration Guide documenting the changes required to successfully run and deploy Fuse 7.11 applications on Camel Spring Boot. It provides information on how to resolve deployment and runtime problems and prevent changes in application behavior. Migration is the first step in moving to the Camel Spring Boot platform. Once the application deploys successfully and runs, users can plan to upgrade individual components to use the new functions and features of Camel Spring Boot.

**Support for EIP circuit breaker**

The Circuit Breaker EIP for Camel Spring Boot supports Resilience4j configuration. This configuration provides integration with Resilience4j to be used as Circuit Breaker in Camel routes.

**Technology Preview extensions**

The following extensions are supported as Technology Preview for CSB 3.20 release version.

- camel-spring-batch-starter
- camel-spring-jdbc-starter
- camel-spring-ldap-starter
- camel-spring-rabbitmq-starter
- camel-spring-redis-starter
- camel-spring-security-starter
- camel-spring-ws-starter

## 5.4. CAMEL SPRING BOOT FIXED ISSUES

The following sections list the issues that have been fixed in Camel Spring Boot.

- Section 5.4.1, “Camel Spring Boot version 3.20.1.1 Fixed Issues”
- Section 5.4.2, “Camel Spring Boot version 3.18.3.1 Fixed Issues”
- Section 5.4.3, “Camel Spring Boot version 3.20 Fixed Issues”
- Section 5.4.4, “Camel Spring Boot version 3.18 Patch 1 Fixed Issues”

### 5.4.1. Camel Spring Boot version 3.20.1.1 Fixed Issues

The following table lists the resolved bugs in Camel Spring Boot version 3.20.1.1.

**Table 5.1. Camel Spring Boot version 3.20.1.1 Resolved Bugs**

| Issue | Description |
|---|---|
| CSB-1524 | CVE-2022-31690 spring-security-oauth2-client: Privilege Escalation in spring-security-oauth2-client [rhint-camel-spring-boot-3] |
| CSB-1718 | CVE-2023-20883 spring-boot: Spring Boot Welcome Page DoS Vulnerability [rhint-camel-spring-boot-3.20] |
| CSB-1719 | CVE-2023-24815 vertx-web: StaticHandler disclosure of classpath resources on Windows when mounted on a wildcard route [rhint-camel-spring-boot-3.20] |
| CSB-1760 | CXF TrustedAuthorityValidatorTest failure |
| CSB-1821 | Backport CAMEL-19421 - Camel-Jira: Use Files.createTempFile in FileConverter instead of creating File directly |

### 5.4.2. Camel Spring Boot version 3.18.3.1 Fixed Issues

The following table lists the resolved bugs in Camel Spring Boot version 3.18.3.1.

**Table 5.2. Camel Spring Boot version 3.18.3.1 Resolved Bugs**

| Issue | Description |
|---|---|
| CSB-656 | CVE-2022-25857 snakeyaml: Denial of Service due to missing nested depth limitation for collections [rhint-camel-spring-boot-3] |
| CSB-714 | CVE-2022-38752 snakeyaml: Uncaught exception in java.base/java.util.ArrayList.hashCode [rhint-camel-spring-boot-3] |
| CSB-715 | CVE-2022-38751 snakeyaml: Uncaught exception in java.base/java.util.regex.Pattern$Ques.match [rhint-camel-spring-boot-3] |

| Issue | Description |
|---|---|
| CSB-716 | CVE-2022-38750 snakeyaml: Uncaught exception in org.yaml.snakeyaml.constructor.BaseConstructor.constructObject [rhint-camel-spring-boot-3] |
| CSB-717 | CVE-2022-38749 snakeyaml: Uncaught exception in org.yaml.snakeyaml.composer.Composer.composeSequenceNode [rhint-camel-spring-boot-3] |
| CSB-719 | CVE-2022-42003 jackson-databind: deep wrapper array nesting wrt UNWRAP_SINGLE_VALUE_ARRAYS [rhint-camel-spring-boot-3] |
| CSB-720 | CVE-2022-42004 jackson-databind: use of deeply nested arrays [rhint-camel-spring-boot-3] |
| CSB-1540 | CVE-2023-1370 json-smart: Uncontrolled Resource Consumption vulnerability in json-smart (Resource Exhaustion) [rhint-camel-spring-boot-3.18] |
| CSB-1702 | CVE-2023-1436 jettison: Uncontrolled Recursion in JSONArray [rhint-camel-spring-boot-3] |
| CSB-1703 | CVE-2022-46364 CXF: Apache CXF: SSRF Vulnerability [rhint-camel-spring-boot-3] |
| CSB-1704 | CVE-2022-46363 CXF: Apache CXF: directory listing / code exfiltration [rhint-camel-spring-boot-3] |
| CSB-1705 | CVE-2022-45047 sshd-common: mina-sshd: Java unsafe deserialization vulnerability |
| CSB-1711 | CVE-2022-25857 snakeyaml: Denial of Service due to missing nested depth limitation for collections [rhint-camel-spring-boot-3] |
| CSB-1712 | CVE-2022-41854 dev-java-snakeyaml: dev-java/snakeyaml: DoS via stack overflow [rhint-camel-spring-boot-3] |
| CSB-1713 | CVE-2022-40156 xstream: Xstream to serialise XML data was vulnerable to Denial of Service attacks [rhint-camel-spring-boot-3] |
| CSB-1714 | CVE-2022-40152 woodstox-core: woodstox to serialise XML data was vulnerable to Denial of Service attacks [rhint-camel-spring-boot-3] |
| CSB-1717 | CVE-2023-20883 spring-boot: Spring Boot Welcome Page DoS Vulnerability [rhint-camel-spring-boot-3.18] |

| Issue | Description |
|---|---|
| CSB-1732 | CXF test failures after undertow version update 2.2.24.SP1-redhat-00001 |
| CSB-1821 | Backport CAMEL-19421 - Camel-Jira: Use Files.createTempFile in FileConverter instead of creating File directly |
| CSB-1947 | CXF TrustedAuthorityValidatorTest failure |

### 5.4.3. Camel Spring Boot version 3.20 Fixed Issues

The following table lists the resolved bugs in Camel Spring Boot version 3.20.

**Table 5.3. Camel Spring Boot version 3.20 Resolved Bugs**

| Issue | Description |
|---|---|
| CSB-656 | CVE-2022-25857 snakeyaml: Denial of Service due to missing nested depth limitation for collections [rhint-camel-spring-boot-3] |
| CSB-699 | CVE-2022-40156 xstream: Xstream to serialise XML data was vulnerable to Denial of Service attacks [rhint-camel-spring-boot-3] |
| CSB-702 | CVE-2022-40152 woodstox-core: woodstox to serialise XML data was vulnerable to Denial of Service attacks [rhint-camel-spring-boot-3] |
| CSB-703 | CVE-2022-40151 xstream: Xstream to serialise XML data was vulnerable to Denial of Service attacks [rhint-camel-spring-boot-3] |
| CSB-714 | CVE-2022-38752 snakeyaml: Uncaught exception in java.base/java.util.ArrayList.hashCode [rhint-camel-spring-boot-3] |
| CSB-715 | CVE-2022-38751 snakeyaml: Uncaught exception in java.base/java.util.regex.Pattern$Ques.match [rhint-camel-spring-boot-3] |
| CSB-716 | CVE-2022-38750 snakeyaml: Uncaught exception in org.yaml.snakeyaml.constructor.BaseConstructor.constructObject [rhint-camel-spring-boot-3] |
| CSB-717 | CVE-2022-38749 snakeyaml: Uncaught exception in org.yaml.snakeyaml.composer.Composer.composeSequenceNode [rhint-camel-spring-boot-3] |
| CSB-719 | CVE-2022-42003 jackson-databind: deep wrapper array nesting wrt UNWRAP_SINGLE_VALUE_ARRAYS [rhint-camel-spring-boot-3] |
| CSB-720 | CVE-2022-42004 jackson-databind: use of deeply nested arrays [rhint-camel-spring-boot-3] |

| Issue | Description |
|---|---|
| CSB-721 | CVE-2022-41852 JXPath: untrusted XPath expressions may lead to RCE attack [rhint-camel-spring-boot-3] |
| CSB-722 | CVE-2022-41853 hsqldb: Untrusted input may lead to RCE attack [rhint-camel-spring-boot-3] |
| CSB-751 | CVE-2022-33681 org.apache.pulsar-pulsar-client: Apache Pulsar: Improper Hostname Verification in Java Client and Proxy can expose authentication data via MITM [rhint-camel-spring-boot-3] |
| CSB-794 | CVE-2022-40150 jettison: memory exhaustion via user-supplied XML or JSON data [rhint-camel-spring-boot-3] |
| CSB-811 | CVE-2022-39368 scandium: Failing DTLS handshakes may cause throttling to block processing of records [rhint-camel-spring-boot-3] |
| CSB-813 | CVE-2022-31777 apache-spark: XSS vulnerability in log viewer UI Javascript [rhint-camel-spring-boot-3] |
| CSB-819 | camel-kafka-starter: KafkaConsumerHealthCheckIT is not working |
| CSB-820 | l2x6 cq-maven-plugin setting wrong version for camel-avro-rpc-component |
| CSB-851 | camel-cxf-rest-starter: EchoService is not an interface error on JDK 17 |
| CSB-852 | camel-infinispan-starter : tests fail on FIPS enabled environment |
| CSB-883 | CVE-2022-37866 apache-ivy: : Apache Ivy: Ivy Path traversal [rhint-camel-spring-boot-3] |
| CSB-904 | CVE-2022-41881 codec-haproxy: HAProxyMessageDecoder Stack Exhaustion DoS [rhint-camel-spring-boot-3] |
| CSB-905 | CVE-2022-41854 dev-java-snakeyaml: dev-java/snakeyaml: DoS via stack overflow [rhint-camel-spring-boot-3] |
| CSB-906 | [archetype] OMP version in openshift profile |
| CSB-929 | CVE-2022-38648 batik: Server-Side Request Forgery [rhint-camel-spring-boot-3] |
| CSB-930 | CVE-2022-38398 batik: Server-Side Request Forgery [rhint-camel-spring-boot-3] |
| CSB-931 | CVE-2022-40146 batik: Server-Side Request Forgery (SSRF) vulnerability [rhint-camel-spring-boot-3] |

| Issue | Description |
|---|---|
| CSB-942 | CVE-2022-4492 undertow: Server identity in https connection is not checked by the undertow client [rhint-camel-spring-boot-3] |
| CSB-1203 | CVE-2022-45047 sshd-common: mina-sshd: Java unsafe deserialization vulnerability |
| CSB-1239 | SAP quickstart spring-boot examples have circular references |
| CSB-1242 | The camel-salesforce-maven-plugin:3.20.1 fails when running with openJDK11 in FIPS mode |
| CSB-1274 | CVE-2021-37533 apache-commons-net: FTP client trusts the host from PASV response by default [rhint-camel-spring-boot-3] |
| CSB-1334 | CVE-2023-24998 tomcat: Apache Commons FileUpload: FileUpload DoS with excessive parts [rhint-camel-spring-boot-3] |
| CSB-1335 | CVE-2022-41966 xstream: Denial of Service by injecting recursive collections or maps based on element’s hash values raising a stack overflow [rhint-camel-spring-boot-3] |
| CSB-1373 | FIPS-mode: Invalid algorythms & security issues on some camel components |
| CSB-1404 | The Spring Boot version is wrong in the BOM |
| CSB-1436 | CVE-2023-20860 springframework: Security Bypass With Un-Prefixed Double Wildcard Pattern [rhint-camel-spring-boot-3] |
| CSB-1437 | CVE-2023-20861 springframework: Spring Expression DoS Vulnerability [rhint-camel-spring-boot-3] |
| CSB-1441 | CVE-2022-42890 batik: Untrusted code execution in Apache XML Graphics Batik [rhint-camel-spring-boot-3] |
| CSB-1442 | CVE-2022-41704 batik: Apache XML Graphics Batik vulnerable to code execution via SVG [rhint-camel-spring-boot-3] |
| CSB-1443 | CVE-2022-37865 apache-ivy: Directory Traversal [rhint-camel-spring-boot-3] |
| CSB-1444 | CVE-2023-22602 shiro-core: shiro: Authentication bypass through a specially crafted HTTP request [rhint-camel-spring-boot-3] |
| CSB-1482 | CVE-2023-1436 jettison: Uncontrolled Recursion in JSONArray [rhint-camel-spring-boot-3] |

| Issue | Description |
|---|---|
| CSB-1499 | Classes generated by camel-openapi-rest-dsl-generator are not added to jar |
| CSB-1533 | [cxfrs-component] camel-cxf-rest-starter needs cxf-spring-boot-autoconfigure |
| CSB-1536 | CVE-2023-20863 springframework: Spring Expression DoS Vulnerability [rhint-camel-spring-boot-3.14] |
| CSB-1540 | CVE-2023-1370 json-smart: Uncontrolled Resource Consumption vulnerability in json-smart (Resource Exhaustion) [rhint-camel-spring-boot-3.18] |

### 5.4.4. Camel Spring Boot version 3.18 Patch 1 Fixed Issues

The following table lists the resolved bugs in Camel Spring Boot version 3.18 Patch 1

**Table 5.4. Camel Spring Boot version 3.18 Patch 1 Resolved Bugs**

| Issue | Description |
|---|---|
| CSB-1537 | CVE-2023-20863 springframework: Spring Expression DoS Vulnerability [rhint-camel-spring-boot-3.18] |
| CSB-1539 | CVE-2023-1370 json-smart: Uncontrolled Resource Consumption vulnerability in json-smart (Resource Exhaustion) [rhint-camel-spring-boot-3.14] |

## 5.5. ADDITIONAL RESOURCES

- Supported Configurations
- Camel Spring Boot Reference
- Getting Started with Camel Spring Boot
- Migration Guide

# CHAPTER 6. SERVICE REGISTRY RELEASE NOTES

Service Registry 2.4 is provided as a General Availability release. Service Registry is a data store for standard event schemas and API designs, and is based on the Apicurio Registry open source community project.

NOTE

Red Hat build of Apicurio Registry is now available as part of Red Hat Application Foundations. Red Hat build of Apicurio Registry 2.x and Red Hat Integration Service Registry 2.x are functionally identical. For more information, see https://access.redhat.com/products/red-hat-application-foundations/.

You can use Service Registry to manage and share the structure of your data using a web console, REST API, Maven plug-in, or Java client. For example, client applications can dynamically push or pull the latest schema updates to or from Service Registry without needing to redeploy. You can also create optional rules to govern how Service Registry content evolves over time. These rules include validation of content, integrity of artifact references, and backwards or forwards compatibility of schema or API versions.

## 6.1. SERVICE REGISTRY INSTALLATION OPTIONS

You can install Service Registry on OpenShift with either of the following data storage options:

- PostgreSQL database
- Red Hat AMQ Streams

For more details, see Installing and deploying Service Registry on OpenShift .

## 6.2. SERVICE REGISTRY SUPPORTED PLATFORMS

Service Registry 2.4 supports the following platforms:

- Red Hat OpenShift Container Platform 4.13, 4.12, 4.11, 4.10
- Red Hat OpenShift Service on AWS
- Microsoft Azure Red Hat OpenShift
- PostgreSQL 15, 14, 13, 12
- Red Hat AMQ Streams 2.4, 2.3, 2.2, 2.1
- OpenJDK 17, 11

For more details, see the following article:

- Red Hat Integration Service Registry Supported Configurations .

Service Registry 2.4 also supports integration with the following products:

- Red Hat Single Sign-On (RH-SSO) 7.6
- Red Hat build of Debezium 2.1

## 6.3. SERVICE REGISTRY NEW FEATURES

Service Registry 2.4 includes the following new features:

### Service Registry core new features

#### Artifact references improvements

- *Artifact reference integrity rule*: You can now apply a new artifact-specific rule and global rule to enforce the integrity of artifact references when creating or updating artifacts. For all artifact types, the rule detects duplicate artifact references and prevents references to nonexistent artifacts. For a subset of artifact types (Apache Avro, Google Protobuf, OpenAPI, and AsyncAPI), the rule ensures that all artifact references are mapped.

- *Automatic artifact reference detection in Maven plug-in*: For a subset of artifact types (Apache Avro, Google Protobuf, and JSON Schema), the Maven plug-in can now automatically identify and configure all of the artifact references for a given artifact when creating or updating artifacts.

- *Visualization of artifact references in the web console* : You can now view both inbound and outbound references for an artifact version, in a new **References** tab. The REST API now supports the retrieval of both inbound and outbound references. Previously, the REST API retrieved outbound references only, and the web console did not display references.

- *Artifact references are now treated as content* : This release now considers artifact references when calculating the content hash and ID, and when checking compatibility of an artifact version. If you upload the same schema content with different references, you create a new artifact version.

#### Client SDK generation (OpenAPI)

This release adds a new web console feature to enable users to generate a client SDK from an OpenAPI artifact. This feature uses Kiota from Microsoft to generate the SDK. The feature runs in your browser only, and cannot be automated by using an API. For more information, see https://github.com/microsoft/kiota.

#### Artifact version deletion

This release adds a new REST API operation, and a new **Delete artifact version** setting in the web console, to enable you to delete artifact versions by using the REST API. Previously, artifact versions were immutable; you could deprecate or disable them, but not delete them. However, it is sometimes necessary to delete an artifact version, despite the semantic implications. For example, you might need to delete an artifact version for regulatory or policy reasons.

#### Core Registry REST API improvements

- *Version comments API*: You can now use the REST API to add comments to artifact versions. Managing comments is not yet available in the web console.

- *Export API supports application/json in the Accept header*: You can now send **application/json** as the value of the **Accept** header in a call to the **/admin/export** API endpoint. Previously, the only way to return an **application/json** formatted response was by using the **forBrowser** query parameter. Now you can use either the query parameter or the **Accept** header.

- *Group management*: You can now optionally manage artifact groups directly by using the REST API, rather than implicitly as part of artifact creation.

#### Confluent Compatibility REST API improvements

- *Updated support for Confluent Compatibility API*: Added support for version 7 of the Confluent Schema Registry API. Service Registry now supports both v6 and v7, at different endpoints.

- *Customizable maximum number of Subjects*: You can now use the **registry.ccompat.max-subjects** dynamic configuration property to customize the maximum number of Subjects returned by the **ccompat** API.

### Other changes

- *Serializer/deserializer artifact resolution with content hash*: **SerDe** classes can now resolve artifact coordinates by using the content hash of the schema.

- *Maven plug-in option to minify Avro*: When registering an Avro schema by using the Maven plug-in, you can now minify the content prior to registration.

### Community-only: Custom artifact types

Users can potentially extend Service Registry with their own custom artifact types, and remove support for existing types. This feature is available only in the community version of Service Registry, and is not supported.

NOTE

To provide this feature, the **ArtifactType** in the REST API was changed from an **enum** to a simple **string**. If you use the REST API client for custom integrations, you might have to edit your code after you upgrade to the newer client.

## Service Registry Operator new features

### Operator support for HTTPS

Added support for configuring HTTPS connections inside the OpenShift cluster. You can create a Secret containing the certificate and the private key, named **tls.crt** and **tls.key** respectively, and reference the Secret by setting the **spec.configuration.security.https.secretName** field in the **ApicurioRegistry** custom resource definition (CRD) file. The Service Registry Operator can then configure an HTTPS port on the Service Registry **Service** resource. If HTTPS is enabled, you can disable the HTTP connection by setting **spec.security.https.disableHttp** to **true**.

### Support for manually managed OpenShift resources

You can now disable certain resource types to ensure that the Service Registry Operator does not create or manage those resources, so that you can configure them manually. Manual configuration provides you with greater flexibility in using features that the Service Registry Operator does not currently support. If you disable a resource type, its existing instance is deleted. If you enable a resource type, the Service Registry Operator attempts to find a resource using the app label (for example, **app=example-apicurioregistry**) and start managing it. Otherwise, the Service Registry Operator creates a new instance. You can disable the following resource types in this way:

- **Ingress**

- **NetworkPolicy**

- **PodDisruptionBudget**

### Improved log level configuration

You can now set the log level for Service Registry and the Service Registry Operator more easily, by using the **spec.configuration.registryLogLevel** field in the **ApicurioRegistry** CRD file. This new field sets the log level for Apicurio application components (excludes non-Apicurio components and libraries), in contrast with the **spec.configuration.logLevel** field, which works for non-Apicurio components and libraries. You can now also set the log level for the Service Registry Operator, by setting the **LOG_LEVEL** environment variable in the Operator **Deployment** resource. Valid **LOG_LEVEL** values are **debug**, **info**, **warn**, and **error**.

### CORS allowed origins

Servers can use the Cross-Origin Resource Sharing (CORS) mechanism to control whether a response can be shared with the origin of the request. The Service Registry Operator now sets the **CORS_ALLOWED_ORIGINS** environment variable, based on the value of the **spec.deployment.host** field in the **ApicurioRegistry** CRD file. This environment variable controls the **Access-Control-Allow-Origin** header sent by Service Registry.

If you use a custom **Ingress** (for example, to configure HTTPS), you can disable the Operator-managed **Ingress** by setting the **spec.deployment.managedResources.disableIngress** field to **true**, and still set the **spec.deployment.host** field to the appropriate value. If you want to configure a fully customized CORS policy, you can set the **spec.deployment.host** field to be empty, apply the changes, and then use the **spec.deployment.env** field to set the **CORS_ALLOWED_ORIGINS** environment variable manually.

### Configuring Service Registry deployment by using a pod template

The **ApicurioRegistry** CRD file now contains the **spec.deployment.podTemplateSpecPreview** field as a Technology Preview feature. The **spec.deployment.podTemplateSpecPreview** field has the same structure as the **spec.template** field in an OpenShift **Deployment** resource (the **PodTemplateSpec** struct). With some restrictions, the Service Registry Operator forwards the data from this field to the corresponding field in the Service Registry **Deployment** resource. This feature provides greater configuration flexibility, without requiring the Service Registry Operator to natively support each use case.

IMPORTANT

Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

### Service Registry user documentation and examples

The documentation library has been updated with the new features available in version 2.4:

- Installing and deploying Service Registry on OpenShift
- Migrating Service Registry deployments
- Service Registry User Guide
- Apicurio Registry v2 core REST API documentation

The open source demonstration applications have also been updated:

- https://github.com/Apicurio/apicurio-registry-examples

## 6.4. SERVICE REGISTRY DEPRECATED FEATURES

### Service Registry core deprecated features

- *Confluent Schema Registry API version 6 (compatibility API)* : Service Registry currently supports two versions of the Confluent Schema Registry API on separate endpoints: version 6 and version 7. Use of version 6 is deprecated; the v6 API endpoint will be removed in a future release. Ensure that you replace all references to the v6 endpoint with references to the v7 API endpoint.

- *Service Registry Core API version 1* : Service Registry support for the original version of the Service Registry Core API (version 1) is now deprecated; the legacy API will be removed in the next major release.

- *Dynamic log level configuration*: The **/admin/loggers** and **/admin/loggers/{logger}** API endpoints are now deprecated in the Service Registry Core API (v2). These endpoints will be removed in a future release.

- *Registry V1 export utility* : Service Registry support for the command-line export utility is now deprecated. The export tool, which is used to export data from Service Registry 1.x into a format that can be imported into 2.x, will no longer be released or maintained. All customers should have already upgraded from 1.x to 2.x.

### Service Registry Operator deprecated features

- *Setting environment variables by editing the Deployment resource* : In previous versions, you could set environment variables for Service Registry by directly editing its **Deployment** resource, which was supported by the Service Registry Operator. Now that you can manage environment variables by using the **spec.configuration.env** field in the **ApicurioRegistry** CRD file, the previous procedure is deprecated and the Operator support for it will be removed. Ensure that you use the **spec.configuration.env** field to set all environment variables that are not set by the Operator.

- *Retention of environment variables for features that are not enabled* : The Service Registry Operator sets environment variables to enable and configure various features, such as Salted Challenge Response Authentication Mechanism (SCRAM) security when using KafkaSQL storage. When such features are disabled, the Operator currently retains the associated environment variables, which can cause problems. Retention of such environment variables is deprecated, and the Operator support for it will be removed. Ensure that your deployment does not rely on the retention of such environment variables.

- *Environment variable precedence* : The Service Registry Operator might attempt to set an environment variable that is already explicitly specified in the **spec.configuration.env** field. If an environment variable has a conflicting value, the value set by the Service Registry Operator takes precedence by default. This behavior will change in the future, to enable users to overwrite most environment variables set by the Operator. Ensure that your deployment does not rely on the original precedence behavior.

## 6.5. UPGRADING AND MIGRATING SERVICE REGISTRY DEPLOYMENTS

You can upgrade the Service Registry server automatically from Service Registry 2.x to Service Registry 2.4 on OpenShift. There is no automatic upgrade from Service Registry 1.x to Service Registry 2.x, and a migration process is required.

### 6.5.1. Updating 2.x client dependencies

It is not mandatory to update client dependencies for this release; existing 2.x clients still work with Service Registry 2.4.

However, before the next release of Service Registry, you must update all of your client application dependencies to use the latest Service Registry client version. Client application dependencies include dependencies for Kafka serializers/deserializers (SerDes), Maven plug-in, and Java REST client. For example, to update the Maven dependencies for a Java REST client, specify the version in your **pom.xml** file as follows:

```
<dependency>
   <groupId>io.apicurio</groupId>
   <artifactId>apicurio-registry-client</artifactId>
   <version>2.4.3.Final-redhat-00006</version>
</dependency>
```

For more details, see Legacy REST API date formats enabled by default .

### 6.5.2. Upgrading from Service Registry 2.x on OpenShift

You can upgrade from Service Registry 2.x on OpenShift 4.9 to Service Registry 2.4 on OpenShift 4.10 or later. You must upgrade both your Service Registry and your OpenShift versions, and upgrade OpenShift one minor version at a time.

**Prerequisites**

- You already have Service Registry 2.x installed on OpenShift 4.9.

**Procedure**

1. In the OpenShift Container Platform web console, click **Administration** and then **Cluster Settings**.
2. Click the pencil icon next to the **Channel** field, and select the next minor **candidate** version (for example, change from **stable-4.9** to **candidate-4.10**).
3. Click **Save** and then **Update**, and wait until the upgrade is complete.
4. If the OpenShift version is less than 4.11, repeat steps 2 and 3, and select **candidate-4.11** or later.
5. Click **Operators** > **Installed Operators** > **Red Hat Integration - Service Registry**.
6. Ensure that the **Update channel** is set to **2.x**.
7. If the **Update approval** is set to **Automatic**, the upgrade should be approved and installed immediately after the **2.x** channel is set.
8. If the **Update approval** is set to **Manual**, click **Install**.
9. Wait until the Operator is deployed and the Service Registry pod is deployed.
10. Verify that your Service Registry system is up and running.

#### Additional resources

- For more details on how to set the Operator update channel in the OpenShift Container Platform web console, see Changing the update channel for an Operator .

### 6.5.3. Migrating from Service Registry 1.1 on OpenShift

For details on migrating from Service Registry 1.1 to Service Registry 2.x, see Migrating Service Registry deployments.

## 6.6. SERVICE REGISTRY RESOLVED ISSUES

**Table 6.1. Service Registry resolved issues in version 2.4.3**

| Issue | Description |
|---|---|
| Registry-3307 | Enabling **Anonymous read access** and **Artifact owner-only authorization** in web console fails with error. |
| Registry-3260 | Key password not optional when configuring KafkaSQL storage to use SSL. |
| Registry-3184 | JSON schema compatibility check fails for **multipleOf** comparison. |
| Registry-3143 | **DownloadReaper** error when deploying Service Registry. |
| Registry-3096 | **SearchedGroup.createdBy** property should have type**String** not **Object**. |
| Registry-3088 | Serialization error in **enum** artifact reference and record ID strategy. |
| Registry-3086 | Cannot upload JSON schemas that reference other schemas with **$ref**. |
| Registry-3080 | Compatibility check does not throw exception when empty schema is provided. |
| Registry-3014 | Upload fails with 422 error when schema has default value of **{}**. |
| Registry-2991 | Kafka consumer thread starts before internal database is ready. |
| Registry-2955 | Redirect filter configured incorrectly. |
| Registry-2952 | Compatibility rules do not order by version number. |
| Registry-2938 | SSL configuration does not work unless SASL is enabled. |
| Registry-2902 | 'registry.ccompat.use-canonical-hash' setting changes schema to canonized form on save. |
| Registry-2880 | Exception not thrown when wrong credentials passed to auth server. |
| Registry-2877 | Uploading new version of Protobuf schema fails with **NullPointerException**. |

| Issue | Description |
|---|---|
| Registry-2826 | Name and description populated incorrectly when creating artifact with REST API. |
| Registry-2790 | Cannot get latest enabled version of schema if latest version is disabled. |
| Registry-2552 | JSON schema validation fails for complex array objects. |

### Service Registry Operator resolved issues

| Issue | Description |
|---|---|
| IPT-916 | Port name not provided in service created by Operator. |
| IPT-883 | Missing environment variables after deployment destroyed. |
| IPT-852 | Incorrect method documented for managing environment variables. |
| Operator-119 | Log level not set correctly. |

## 6.7. SERVICE REGISTRY RESOLVED CVES

Table 6.2. Service Registry resolved Common Vulnerabilities and Exposures (CVEs) in version 2.4.3

| Issue | Description |
|---|---|
| IPT-943 | CVE-2023-2976 guava: Insecure temporary directory creation [rhint-serv-2]. |
| IPT-890 | CVE-2021-46877 jackson-databind: Possible DoS if using JDK serialization to serialize JsonNode [rhint-serv-2]. |
| IPT-880 | CVE-2022-3510 protobuf-java: Message-Type Extensions parsing issue leads to DoS [rhint-serv-2]. |
| IPT-879 | CVE-2022-3509 protobuf-java: Textformat parsing issue leads to DoS [rhint- serv-2]. |
| IPT-876 | CVE-2023-28867 graphql-java: crafted GraphQL query causes stack consumption [rhint-serv-2]. |
| IPT-866 | CVE-2022-25881 http-cache-semantics: Regular Expression Denial of Service (ReDoS) vulnerability [rhint-serv-2]. |
| IPT-856 | CVE-2022-3782 keycloak: path traversal via double URL encoding [rhint-serv-2]. |
| IPT-850 | CVE-2022-45787 apache-james-mime4j: Temporary File Information Disclosure in MIME4J TempFileStorageProvider [rhint-serv-2]. |

| Issue | Description |
|---|---|
| IPT-845 | CVE-2022-4742 json-pointer: prototype pollution in json-pointer [rhint-serv-2]. |
| IPT-825 | CVE-2022-40152 woodstox-core: woodstox to serialise XML data was vulnerable to Denial of Service (DoS) attacks [rhint-serv-2]. |

## 6.8. SERVICE REGISTRY KNOWN ISSUES

The following known issues apply in Service Registry 2.4.x:

### Service Registry core known issues

#### Registry-3413 - Legacy REST API date formats enabled by default

For maximum compatibility and for easier upgrades from older versions of Service Registry, the date format used in the REST API is not compliant with OpenAPI standards (because of a bug in older versions).

Before the next release of Service Registry, you must upgrade all of your client applications to use the latest client version. The next release will fix the date format bug, which will result in older clients no longer being compatible with the REST API.

To update your REST API to be OpenAPI compliant, you can fix the date format bug in *this* version of Service Registry as follows:

1. Update all of your client applications to version **2.4.3.Final-redhat-00006**, as described in Updating 2.x client dependencies.

2. Set the following environment variable to the value shown:

```
REGISTRY_APIS_V2_DATE_FORMAT=yyyy-MM-dd'T'HH:mm:ss'Z'
```

#### IPT-814 - Service Registry logout feature incompatible with RH-SSO 7.6

In RH-SSO 7.6, the **redirect_uri** parameter used with the logout endpoint is deprecated. For more details, see the RH-SSO 7.6 Upgrading Guide. Because of this deprecation, when Service Registry is secured by using the RH-SSO Operator, clicking the **Logout** button displays the **Invalid parameter: redirect_uri** error.

For a workaround, see https://access.redhat.com/solutions/6980926.

#### IPT-701 - CVE-2022-23221 H2 allows loading custom classes from remote servers through JNDI

When Service Registry data is stored in AMQ Streams, the H2 database console allows remote attackers to execute arbitrary code by using the JDBC URL. Service Registry is not vulnerable by default and a malicious configuration change is required.

### Service Registry Operator known issues

#### Operator-42 - Autogeneration of OpenShift route might use wrong base host value

If multiple **routerCanonicalHostname** values are specified, autogeneration of the Service Registry OpenShift route might use a wrong base host value.

# APPENDIX A. USING YOUR SUBSCRIPTION

Integration is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

## ACCESSING YOUR ACCOUNT

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

## ACTIVATING A SUBSCRIPTION

1. Go to access.redhat.com.
2. Navigate to **My Subscriptions**.
3. Navigate to **Activate a subscription** and enter your 16-digit activation number.

## DOWNLOADING ZIP AND TAR FILES

To access zip or tar files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal **Product Downloads** page at access.redhat.com/downloads.
2. Scroll down to **INTEGRATION AND AUTOMATION**.
3. Click **Red Hat Integration** to display the Red Hat Integration downloads page.
4. Click the **Download** link for your component.

*Revised on 2023-09-22 08:24:28 UTC*