



# Red Hat Integration 2021.Q1

## Release Notes for Red Hat Integration 2021.Q1

What's new in Red Hat Integration



# Red Hat Integration 2021.Q1 Release Notes for Red Hat Integration 2021.Q1

---

What's new in Red Hat Integration

## Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

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

---

## Table of Contents

<b>CHAPTER 1. RED HAT INTEGRATION</b> .....	<b>3</b>
<b>CHAPTER 2. NEW FEATURES IN THIS RELEASE</b> .....	<b>4</b>
2.1. NEW INTEGRATION FEATURES	4
2.2. NEW COMPONENT FEATURES	4
<b>CHAPTER 3. DEBEZIUM RELEASE NOTES</b> .....	<b>5</b>
3.1. DEBEZIUM DATABASE CONNECTORS	5
3.2. DEBEZIUM SUPPORTED CONFIGURATIONS	6
3.3. DEBEZIUM INSTALLATION OPTIONS	6
3.4. NEW DEBEZIUM FEATURES	6
<b>CHAPTER 4. CAMEL K RELEASE NOTES</b> .....	<b>8</b>
4.1. NEW CAMEL K FEATURES	8
4.1.1. Platform and core component versions	8
4.1.2. Technology Preview features	8
4.1.3. Camel K Operator metadata	9
4.2. CAMEL K KNOWN ISSUES	9
4.3. CAMEL K FIXED ISSUES	10
4.3.1. Fixed CVE security issues	10
<b>CHAPTER 5. CAMEL KAFKA CONNECTOR RELEASE NOTES</b> .....	<b>11</b>
5.1. NEW CAMEL KAFKA CONNECTOR FEATURES	11
5.1.1. Platform and core component versions	11
5.1.2. Technology Preview features	11
5.1.3. Technology Preview Camel Kafka connectors	12
5.2. CAMEL KAFKA CONNECTOR FIXED ISSUES	13
5.2.1. Fixed CVE security issues	13
<b>CHAPTER 6. RED HAT INTEGRATION OPERATORS</b> .....	<b>14</b>
6.1. WHAT OPERATORS ARE	14
6.2. RED HAT INTEGRATION OPERATOR	14
6.2.1. Support life cycle	15
6.3. RED HAT INTEGRATION COMPONENT OPERATORS	15
6.3.1. 3scale Operators	15
6.3.2. AMQ Operators	15
6.3.3. Camel K Operator	15
6.3.4. Fuse Operators	15
6.3.5. Service Registry Operator	15



# 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:

- API connectivity
- Data transformation
- Service composition and orchestration
- Real-time messaging
- Cross-datacenter message streaming
- API management

## Additional resources

- [Understanding enterprise integration](#)

## CHAPTER 2. NEW FEATURES IN THIS RELEASE

This section provides a summary of the key new features in Red Hat Integration 2021.Q1 and provides links to more details on new features available in different components.



### NOTE

These release notes include details on components updated in Red Hat Integration 2021.Q1 only. For details on the latest versions of other components, such as Service Registry, see [Red Hat Integration Release Notes for 2020-Q4](#) .

## 2.1. NEW INTEGRATION FEATURES

### Red Hat Integration Operator

- Install and upgrade the OpenShift Operators that manage your Red Hat Integration components with [Red Hat Integration Operator 1.0](#)

### Data integration

- Change data capture and real-time events, including new DB2 connector and integration with Service Registry in [Debezium 1.4](#)

### Serverless Camel K

- Cloud-native integration for serverless architectures based on Camel 3.7 with [Red Hat Integration - Camel K \(Technology Preview\)](#)

### Event-driven Camel Kafka connectors

- Camel components available as Kafka connectors to external systems with [Camel Kafka Connector \(Technology Preview\)](#)

## 2.2. NEW COMPONENT FEATURES

For more details on what's new in Red Hat Integration 2021.Q1 components:

- Red Hat 3scale API Management
  - [Red Hat 3scale API Management 2.9 On-Premises Release Notes](#)
  - [Red Hat 3scale API Management SaaS Release Notes](#)
- [Red Hat AMQ 2021.Q1 Product Documentation](#)
- [Red Hat Fuse 7.8 Release Notes](#)



## CHAPTER 3. DEBEZIUM RELEASE NOTES

Red Hat Integration 2021.Q1 includes a General Availability release of Debezium on OpenShift based on the [Debezium](#) open source project. Debezium is a distributed change data capture platform that tracks database operations and streams data change events. Debezium is built on Apache Kafka and is deployed and integrated with AMQ Streams.

Debezium captures row-level changes to database tables and passes corresponding change event records to AMQ Streams. Applications can read these *change event streams* and access the change events in the order in which they occurred.

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, “New Debezium features”](#)

### 3.1. DEBEZIUM DATABASE CONNECTORS

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

- Db2
- MongoDB
- MySQL
- PostgreSQL
- SQL Server



#### NOTE

- 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.
- Currently, you cannot use the transaction metadata feature of the Debezium MongoDB connector with MongoDB 4.2.
- The Debezium PostgreSQL connector requires you to 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](#)

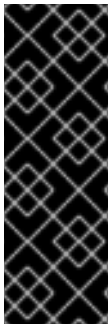
## 3.2. DEBEZIUM SUPPORTED CONFIGURATIONS

For information about Debezium supported configurations, including information about supported database versions, see the [Debezium 1.4 Supported configurations page](#).

## 3.3. DEBEZIUM INSTALLATION OPTIONS

You can install Debezium with AMQ Streams on OpenShift or RHEL:

- [Installing Debezium on OpenShift - General Availability](#)
- [Installing Debezium on RHEL - Technology Preview](#)



### 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. 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](#).

## 3.4. NEW DEBEZIUM FEATURES

Debezium 1.4 includes the following updates:

### Promoted to GA

The following features that were offered as Technology Previews in the previous release are now available for General Availability:

#### [Debezium Db2 connector](#)

Connector for IBM Db2 database (LUW)

#### [Content-based router](#)

Single message transformation (SMT) for re-routing data change event records to topics based on event content.

#### [Filter SMT](#)

Evaluates expressions for each change event and drops or emits the event based on the evaluation result.

#### [Avro serialization](#)

Support for configuring Debezium connectors to use Avro to serialize message keys and values.

### Technology Preview features

#### [CloudEvents converter](#)

Emits change event records that conform to the CloudEvents specification. Avro encoding type is now supported for the CloudEvents envelope structure.

#### [Outbox event router](#)

SMT that supports the outbox pattern for safely and reliably exchanging data between multiple (micro) services.

### Debezium documentation

- Information about how to enable and customize [Kafka Connect automatic topic creation](#).
- Information about how the capture job configurations on [Db2](#) and [SQL Server](#) affect server load and latency.

## CHAPTER 4. CAMEL K RELEASE NOTES

Red Hat Integration - Camel K is available as a Technology Preview component in Red Hat Integration 2021.Q1. 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.



### 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.

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](#).

## 4.1. NEW CAMEL K FEATURES

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

### 4.1.1. Platform and core component versions

- OpenShift Container Platform 4.6 or 4.7
- OpenShift Serverless 1.13
- Quarkus 1.7 Java runtime
- Apache Camel K 1.3
- Apache Camel 3.7
- Apache Camel Quarkus 1.5
- OpenJDK 11

### 4.1.2. Technology Preview 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, XML, or YAML DSL
- Development tooling with Visual Studio Code
- Monitoring of integrations using Prometheus in OpenShift
- Quickstart tutorials, including new Transformations and SaaS
- Kamelet Catalog for source connectors to external systems such as AWS, Jira, and Salesforce

#### Additional resources

- [Deploying Camel K Integrations on OpenShift](#)

### 4.1.3. Camel K Operator metadata

The Camel K Technology Preview 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](#) .

## 4.2. CAMEL K KNOWN ISSUES

The following known issues apply to the Camel K Technology Preview:

### **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-15787** - kamel run command does not work for remote files on Windows

On Windows, the **kamel run** command raises an error when attempting to run remote Camel integration files. For example:

```
> kamel run https://raw.githubusercontent.com/apache/camel-k/b29333f0a878d5d09fb3965be8fe586d77dd95d0/e2e/common/files/yaml.yaml
panic: runtime error: invalid memory address or nil pointer dereference...
```

### **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 Technology Preview and has community-only support.

For more details, see the [Apache Camel K community](#) .

### **ENTESB-15893 - Camel K catalog contains camel-quarkus-spark reference and cannot deploy integrations with Apache Spark**

The Camel K catalog includes the **camel-quarkus-spark** component, which is no longer included in the Bill of Materials (BOM) for Camel Quarkus extensions. When you try to deploy a Camel K integration using the Spark component in Camel Quarkus, the integration cannot be compiled due to this missing dependency.

For more details, see the [Spark component in Camel Quarkus](#).

### **ENTESB-15930 - Camel K dependency autoloading not working correctly with YAML format**

When using the **- route** attribute in Camel integrations written in YAML, Camel K dependency autoloading does not work correctly. However, Camel K dependency autoloading works correctly when using the **- from** attribute instead.

For more details, see the [Apache Camel K community](#).

## **4.3. CAMEL K FIXED ISSUES**

The following issues have been fixed in the Camel K Technology Preview:

### **4.3.1. Fixed CVE security issues**

#### **ENTESB-14997 - CVE-2020-25649 jackson-databind: FasterXML DOMDeserializer insecure entity expansion is vulnerable to XML external entity**

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allowed vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability was data integrity.

## CHAPTER 5. CAMEL KAFKA CONNECTOR RELEASE NOTES

Camel Kafka Connector is available as a Technology Preview component in Red Hat Integration 2021.Q1.

**Note:** There are no plans for a release of Camel Kafka Connector beyond the unsupported Technology Preview release.

Using Camel Kafka Connector, you can configure standard Camel components as connectors in Kafka Connect. This widens the scope of possible integrations beyond the external systems supported by Kafka Connect alone.

Camel Kafka Connector provides a user-friendly way to configure Camel components directly in the Kafka Connect framework. You can leverage Camel components for integration with different systems by connecting to or from Camel Kafka sink or source connectors. You do not need to write any code, and can include the appropriate connector JARs in your Kafka Connect image and configure the connector options using custom resources.

Camel Kafka Connector is built on Apache Camel Kafka Connector, which is a subproject of the Apache Camel open source community. Camel Kafka Connector is fully integrated with OpenShift Container Platform, AMQ Streams, and Kafka Connect. Camel Kafka Connector is available with the Red Hat Integration - Camel K distribution for cloud-native integration on OpenShift.



### 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.

For more information about support scope, see [Technology Preview Features Support Scope](#).

## 5.1. NEW CAMEL KAFKA CONNECTOR FEATURES

The Camel Kafka Connector Technology Preview includes the following main features:

### 5.1.1. Platform and core component versions

- OpenShift Container Platform 4.6 or 4.7
- Red Hat Enterprise Linux 8.x
- AMQ Streams 1.6
- Apache Kafka Connect 2.6
- Apache Camel Kafka Connector 0.7.1
- Apache Camel 3.7
- OpenJDK 11

### 5.1.2. Technology Preview features

- Selected Camel Kafka connectors

- Marshaling/unmarshalling of Camel data formats for sink and source connectors
- Aggregation for sink connectors
- Maven archetypes for extending connectors

### 5.1.3. Technology Preview Camel Kafka connectors

Table 5.1. Camel Kafka connectors in Technology Preview

Connector	Sink/source
Amazon Web Services (AWS2) Kinesis	Sink and source
Amazon Web Services (AWS2) S3	Sink and source
Amazon Web Services (AWS2) SNS	Sink only
Amazon Web Services (AWS2) SQS	Sink and source
Azure Storage Blob	Sink only
Azure Storage Queue	Sink only
Cassandra Query Language (CQL)	Sink and source
Elasticsearch	Sink only
File	Sink only
Hadoop Distributed File System (HDFS)	Sink only
Hypertext Transfer Protocol (HTTP)	Sink only
Java Database Connectivity (JDBC)	Sink only
Java Message Service (JMS)	Sink and source
MongoDB	Sink and source
RabbitMQ	Sink and source
SQL	Sink and source
SSH	Sink and source
Syslog	Sink and source



Connector	Sink/source
Timer	Source only

### Additional resources

- [Getting Started with Camel Kafka Connector](#)

## 5.2. CAMEL KAFKA CONNECTOR FIXED ISSUES

The following issues have been fixed in the Camel Kafka Connector Technology Preview:

### 5.2.1. Fixed CVE security issues

#### **[ENTESB-14997](#) - CVE-2020-25649 jackson-databind: FasterXML DOMDeserializer insecure entity expansion is vulnerable to XML external entity**

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allowed vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability was data integrity.

## CHAPTER 6. RED HAT INTEGRATION OPERATORS

Red Hat Integration provides Operators to automate the deployment of Red Hat Integration components on OpenShift. You can use the Red Hat Integration Operator to manage multiple component Operators. Alternatively, you can manage each component Operator individually. This section introduces Operators and provides links to detailed information on how to use Operators to deploy Red Hat Integration components.

### 6.1. WHAT OPERATORS ARE

Operators are a method of packaging, deploying, and managing a Kubernetes application. They take human operational knowledge and encode it into software that is more easily shared with consumers to automate common or complex tasks.

In OpenShift Container Platform 4.x, the *Operator Lifecycle Manager (OLM)* helps users install, update, and generally manage the life cycle of all Operators and their associated services running across their clusters. It is part of the Operator Framework, an open source toolkit designed to manage Kubernetes native applications (Operators) in an effective, automated, and scalable way.

The OLM runs by default in OpenShift Container Platform 4.x, which aids cluster administrators in installing, upgrading, and granting access to Operators running on their cluster. The OpenShift Container Platform web console provides management screens for cluster administrators to install Operators, as well as grant specific projects access to use the catalog of Operators available on the cluster.

*OperatorHub* is the graphical interface that OpenShift cluster administrators use to discover, install, and upgrade Operators. With one click, these Operators can be pulled from OperatorHub, installed on the cluster, and managed by the OLM, ready for engineering teams to self-service manage the software in development, test, and production environments.

#### Additional resources

- For more information about Operators, see the [OpenShift documentation](#).

### 6.2. RED HAT INTEGRATION OPERATOR

You can use the Red Hat Integration Operator to install and upgrade multiple Red Hat Integration component Operators:

- 3scale
- 3scale APIcast
- AMQ Broker
- AMQ Interconnect
- AMQ Streams
- API Designer
- Camel K
- Fuse Console

- [Fuse Online](#)
- [Service Registry](#)

### 6.2.1. Support life cycle

To remain in a supported configuration, you must deploy the latest Red Hat Integration Operator version. Each Red Hat Integration Operator release version is only supported for 3 months.

#### Additional resources

- For more details on managing multiple Red Hat Integration component Operators, see [Installing the Red Hat Integration Operator on OpenShift](#).

## 6.3. RED HAT INTEGRATION COMPONENT OPERATORS

You can install and upgrade each Red Hat Integration component Operator individually, for example, using the 3scale Operator, the Camel K Operator, and so on.

### 6.3.1. 3scale Operators

- [3scale Operator](#)
- [3scale APIcast Operator](#)

### 6.3.2. AMQ Operators

- [AMQ Broker Operator](#)
- [AMQ Interconnect Operator](#)
- [AMQ Streams Cluster Operator](#)
- [AMQ Online Operator](#)

### 6.3.3. Camel K Operator

- [Camel K Operator \(Technology Preview\)](#)

### 6.3.4. Fuse Operators

- [Fuse on OpenShift - Samples Operator](#)
- [Fuse on OpenShift - Fuse Console Operator](#)
- [Fuse on OpenShift - API Designer Operator](#)
- [Fuse Online Operator](#)

### 6.3.5. Service Registry Operator

- [Service Registry Operator](#)

### Additional resources

- For details on managing multiple Red Hat Integration component Operators, see [Installing the Red Hat Integration Operator on OpenShift](#).