



Red Hat boosts Apache Kafka connectivity for Kubernetes environments with latest Red Hat Integration release

Author: [Sameer Parulkar](#), Product Marketing Director for Red Hat Integration

Enterprises around the globe are reshaping their businesses to respond more quickly to the changes happening around us. [Event driven architectures](#) (EDA) are at the heart of this effort to transition processes and applications to an efficient model. To implement EDA solutions, many enterprises are turning to [Apache Kafka](#).

When a distributed data streaming platform such as Apache Kafka is deployed in a [Kubernetes](#) environment like [Red Hat OpenShift](#), customers can achieve the performance, efficiency and scale they need. Red Hat has focused the latest release of [Red Hat Integration](#) on extending Apache Kafka connectivity to help customers create event-driven applications that integrate into every aspect of their businesses.

Introducing Full Apache Camel K Support

To deliver improved Apache Kafka connectivity, this release of Red Hat Integration includes full support for Apache Camel K, which was [previously available as a technology preview](#). Apache Camel K builds upon the success of the Apache Camel community, which has long been the cornerstone of open source [integration](#). For Red Hat customers, the Camel K capabilities can be used with self-managed Red Hat Integration and also with the fully managed [Red Hat OpenShift Streams for Apache Kafka](#) (developer preview).

With Camel K, we are bringing the breadth of open source Camel source and sink connectors to the Red Hat portfolio, giving customers the ability to integrate everything from SaaS applications to traditional systems and Kafka-based applications, all with greater flexibility to implement and scale data stream processing as part of their overall event driven architectures in whichever manner best suits their unique business needs. By bringing Camel K to the Red Hat Integration portfolio, Red Hat is giving customers a lightweight integration framework that runs natively on Red Hat OpenShift, the industry's leading enterprise Kubernetes platform, and is specifically designed for [serverless](#) and [microservices](#) architectures.

With Red Hat Integration, developers can write [cloud-native](#) services that can interact with Apache Kafka events and execute sophisticated integration logic. Furthermore, Camel K is built on [Quarkus](#), the leading Java framework for Kubernetes and serverless environments. The smaller footprint and faster start-up times of Quarkus means that the integration logic can scale up or down dynamically in response to demand.

New Access to Kamelets

Apache Camel codifies common enterprise integration patterns (EIPs) and hundreds of connectors that can be used to efficiently build and deploy connected applications across the enterprise, and is one of the core open source projects that [Red Hat Fuse](#) is based on. The Camel K project builds on that foundation and provides what we believe is a better and simpler experience for Kafka and Java developers looking to connect applications both inside and outside the enterprise.

One of the ways it does this is through [Kamelets](#), a relatively new component in Camel K that enables developers to declaratively configure interactions with connectors, for example Kafka streams to Salesforce. Kamelets extracts the coding complexity of connection logic and replaces it with visual forms that collect only the necessary information, such as location and security parameters. Since the definitions are encoded in standardized forms like YAML, the forms can be opened and configured with visual tools. Once configured, the connections are available for usage in Camel routes, either graphically or in a domain specific language (DSL).

Kamelets are also community assets. Camel K users can benefit from the ever-expanding list of Kamelets in the Kamelet [catalog](#), which includes connectors to many databases, applications, and SaaS resources.

In addition to Camel K and Kamelets, the latest release of Red Hat Integration introduces several incremental updates to the following components:

- Red Hat AMQ Interconnect 2.0: Based on the Apache Qpid project, AMQ Interconnect is a lightweight AMQP message router for building scalable, available, and performant messaging networks. This AMQ Interconnect version introduces a development preview of a service network capability that links services across hybrid cloud environments. A service network enables communication between Red Hat OpenShift clusters running in different network locations, including private clusters. This enables geographically distributed services to connect as if they were all running in the same site. The service networks can be deployed and managed using the [Skupper](#) command line interface (CLI).
- Debezium 1.5: The Debezium project is a distributed change data platform that captures row-level database changes so that applications can respond immediately. This release of Debezium expands the support for the Red Hat Integration Service Registry, as well as adding support for transaction metadata to MySQL databases. The metadata support means that events emitted by Debezium can include transaction initiations, commits, and rollbacks (this metadata is already available for other database platforms with Debezium).

These components are featured in the next release of Red Hat Integration, which will be generally available in early August. Customers can get the latest updates from the [Red Hat Customer Portal](#). Learn more about Camel K [in this video](#), explore additional considerations for connecting and composing Kubernetes-native applications [in this paper](#), or [get started with Red Hat OpenShift streams for Apache Kafka](#).