5 considerations for connecting and composing Kubernetes-native apps

Apache Camel K is a lightweight cloud-native integration framework that runs natively on Kubernetes. Based on Apache Camel, Camel K is designed and optimized for serverless and microservices architectures. The framework declaratively orchestrates events in a serverless cloud. As a result, developers gain access to more than 300 connectors and built-in integration patterns to connect any application to different applications in a flexible, scalable way. Apache Camel K is part of Red Hat® Integration, a cloud-native integration platform.

This checklist details five key considerations, and related benefits, for using Camel K to connect and compose Kubernetes-native apps.

1 Go cloud-native without the dependency

Specifying dependencies is a mundane part of the job that every developer must deal with. As organizations strive to develop composable apps more quickly, writing long dependency definitions, even for small microservices or simple functions, can consume valuable developer time and stifle delivery.

Camel K automatically detects, manages, and pulls dependencies available to the operator. This approach saves valuable time because it:

- Eliminates the need to determine which libraries must be added.
- Sets and shares dependencies among developers.
- Ensures testing for security is not missed.

2 Simplify application connectivity

At its core, the Camel K app is Apache Camel, which means developers benefit from well-established patterns that can be reused. The result is faster development time because the building blocks are already created–there is no need to start from scratch. Everything is standardized, which makes it easier to customize applications. The key benefits include:

- Well-established patterns to reuse. Prebuilt transformations for data.
- 300+ components available for quick connectivity from Software-as-a-Service (SaaS) applications to data sources.
- Faster development time with modular app builds.

3 Scale serverless infrastructure

Serverless is, by its nature, event-driven. In a serverless architecture, when an app is not in use it will dial down to zero. When needed, Camel K provides the orchestration to "wake up" serverless workloads and let them run in a cloud environment. With Red Hat OpenShift®, Camel K will detect if Red Hat OpenShift Serverless is installed and available to create the services needed. You can:

- Automatically connect to external sources and trigger the functions needed to run the app.
- Run batched jobs or integrations with other systems happening on the backend of the app.
- Build and deploy workloads with Camel K.
- Focus on building code, thanks to operator patterns and underlying serverless technologies.
- Deploy in the form of a serverless function, serverless source or sink, or a long-running microservice.

4 Accelerate template-centric development

A Kamelet, part of Camel K, uses Camel components and enterprise integration patterns (EIPs) to describe a system's behavior using template-based design. You can reuse a Kamelet abstraction in any integration on top of a Kubernetes cluster to achieve the needed communication in a much quicker way.

- Use a Kamelet as a source or sink for events, making them a building block for event-driven architecture.
- Discover and use Kamelets for other applications – or reuse them within your organization.
- Declare the kind of data type that the Kamelet will use as input and produce as output.
- Use any of Camel's domain-specific languages (DSLs) to write a Kamelet, such as YAML DSL.

5 Support event-driven architecture

An event-driven architecture (EDA) approach for designing applications and services helps you respond to real-time information based on the sending and receiving of information about individual event notifications.

Because you are not broadcasting all events to every application in your system all of the time, you require orchestration. Camel K provides the connectors to integrate the events you need to run your apps, when and where they are needed. Camel K includes connectivity with Apache Kafka to process real-time data streams.

- Filter multiple event types to the right location at the right time and mirror real-world needs.
- Manipulate events that are flowing through your system using prebuilt templates.
- Allow developers to immediately update apps they are building on the platform.
- Decouple from time so you can process events that are being used at different times, allowing you to scale and boost elasticity.

Learn more

Learn more about serverless integration with Camel K at red.ht/integration.

Try Serverless Integration with Camel K

Read more about serverless integrations. Find, try, and deploy with flexible software at Red Hat Marketplace.

About Red Hat

Red Hat helps customers standardize across environments, develop cloud-native applications, and integrate, automate, secure, and manage complex environments with award-winning support, training, and consulting services.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

North America
1 888 REDHAT1
www.redhat.com

Europe, Middle East, and Africa
00800 7334 2835
europe@redhat.com

Asia Pacific
+65 6490 4200
apac@redhat.com

Latin America
+54 11 4329 7300
info-latam@redhat.com

redhat.com
#F29541_0721