

What is Apache Kafka?

Apache Kafka is a distributed data streaming platform that can publish, subscribe to, store, and process streams of records in real time. It is designed to handle data streams from multiple sources and deliver them to multiple consumers. In short, it moves massive amounts of data—not just from point A to B, but from points A to Z and anywhere else you need, all at the same time.

Apache Kafka is an alternative to a traditional enterprise messaging system. It started out as an internal system developed by LinkedIn to handle 1.4 trillion messages per day, but now it's an open source data streaming solution with application for a variety of enterprise needs.

Asynchronous integration with Apache Kafka

[Microservices](#) have changed the development landscape. They make developers more agile by reducing dependencies, such as shared database tiers. But the distributed applications your developers are building still need some type of integration to share data. One popular integration option, known as the synchronous method, utilizes application programming interfaces (APIs) to share data between different users.

A second integration option, the asynchronous method, involves replicating data in an intermediate store. This is where Apache Kafka comes in, streaming data from other development teams to populate the data store, so the data can be shared between multiple teams and their applications.

Microservices teams have different requirements for integration than traditional, waterfall development teams. These teams require 3 foundational capabilities:

1. [Distributed integrations](#): Lightweight, patterns-based integrations that can be continuously deployed where required, and are not limited by centralized ESB type deployments.
2. [APIs](#): API-based services to foster an ecosystem of partners, customers, and developers that can offer reliable and profitable use of services.
3. [Containers](#): Platform to develop, manage, and scale [cloud-native](#) and connected applications. Containers enable development of lean artifacts that are individually deployable, part of [DevOps](#) processes, and supported by out-of-box clustering, ensuring high availability.

Red Hat calls this approach "agile integration," which allows integrations to be part of application development processes, providing more agility and adaptive solutions. Part of agile integration is the freedom to use either synchronous or asynchronous integration, depending on the specific needs of the application. Apache Kafka is a great option when using asynchronous event driven integration to augment your use of synchronous integration and APIs, further supporting microservices and enabling agile integration. In this way, Apache Kafka can be an important part of your initiative to streamline the development process, drive innovation, save time, and ultimately speed up time to market for your new features, apps, and services.

When to use Apache Kafka

Apache Kafka is built into streaming data pipelines that share data between systems and/or applications, and it is also built into the systems and applications that consume that data. Apache Kafka supports a range of use cases where high throughput and scalability are vital. Since Apache Kafka minimizes the need for point-to-point integrations for data sharing in certain applications, it can reduce latency to milliseconds. This means data is available to users faster, which can be advantageous in use cases that require real-time data availability, such as IT operations and e-commerce.

Apache Kafka can handle millions of data points per second, which makes it well-suited for [big data](#) challenges. However, Kafka also makes sense for companies that are not currently handling such extreme data scenarios. In many data processing use cases, such as the [Internet of Things](#) (IoT) and social media, data is increasing exponentially, and may quickly overwhelm an application you are building based on today's data volume. In terms of data processing, you must consider scalability, and that means planning for the increased proliferation of your data.

IT operations

IT Operations is all about data. IT Operations needs access to the data, and they need it quickly. This is the only way to keep websites, applications, and systems up and running and performing at all times. Apache Kafka is a good fit for IT Operations functions that rely on collecting data from a variety of sources such as monitoring, alerting, and reporting; log management; and tracking website activity.

Internet of Things

[According to Gartner, IoT is expected to include more than 20 billion devices by 2020.](#) The value of IoT is the actionable data generated by this multitude of sensors. Apache Kafka is designed for scalability to handle the massive amount of data expected from IoT.

E-commerce

E-commerce is a growing opportunity for using Apache Kafka, which can process data such as page clicks, likes, searches, orders, shopping carts, and inventory.

How Kubernetes scales Apache Kafka applications

Kubernetes is the ideal platform for Apache Kafka. Developers need a scalable platform to host Kafka applications, and Kubernetes is the answer.

Like Apache Kafka, Kubernetes also makes your development process more agile. Kubernetes—the technology behind Google’s cloud services—is an open source system for managing containerized applications, and it eliminates many of the manual processes associated with containers. Using Apache Kafka in Kubernetes streamlines the deployment, configuration, management, and use of Apache Kafka.

By combining Kafka and Kubernetes, you gain all the benefits of Kafka, and also the advantages of Kubernetes: scalability, high availability, portability and easy deployment.

The scalability of Kubernetes is a natural complement to Kafka. In Kubernetes, you can scale resources up and down with a simple command, or scale automatically based on usage as needed to make the best use of your computing, networking, and storage infrastructure. This capability enables Apache Kafka to share a limited pool of resources with other applications. Kubernetes also offers Apache Kafka portability across infrastructure providers and operating systems. With Kubernetes, Apache Kafka clusters can span across on-site and public, private, or hybrid clouds, and use different operating systems.

Integrate with Red Hat

Red Hat gives you modular, lightweight, and comprehensive integration solutions that are open source, open standards, and available on-premise or in the cloud.

MESSAGING

RED HAT® AMQ

As a part of Red Hat® AMQ, [AMQ streams](#) offers Apache Kafka on Red Hat OpenShift® Container Platform. AMQ streams is a scalable, distributed, and high-performance data streaming capability based on the Apache Kafka project, running on Kubernetes.

[Learn more →](#)

CONTAINER PLATFORM

RED HAT® OPENSIFT

Develop, deploy, manage, and scale cloud-native applications with containers through a container platform. OpenShift can orchestrate all of those applications automatically and work with offerings like AMQ, Fuse, and 3scale API Management to bring an agile integration approach to the cloud.

[Learn more →](#)

Keep exploring integration with Red Hat

Try It

Learn more

Talk to a Red Hatter