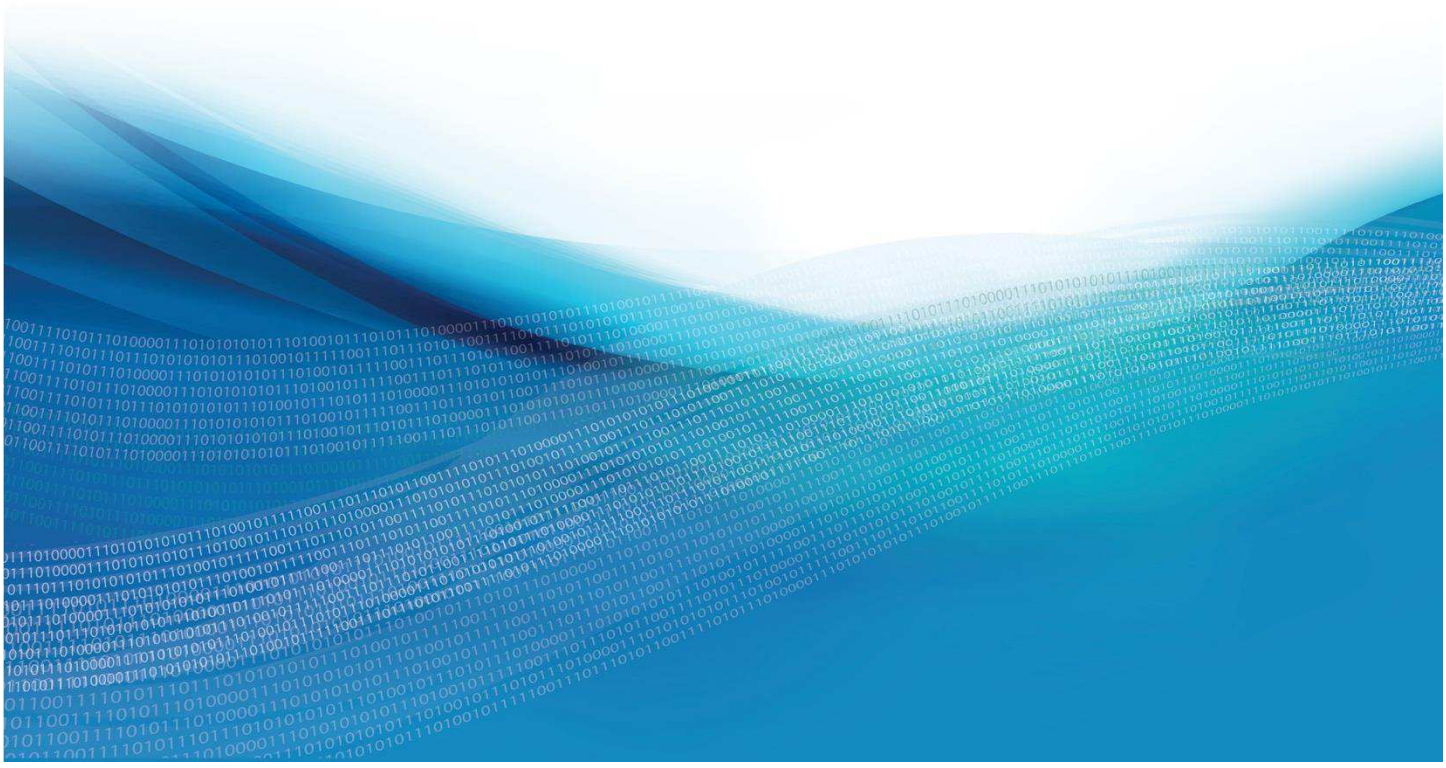
5 Ways StreamSets Tames Apache Kafka®



StreamSets

Introduction

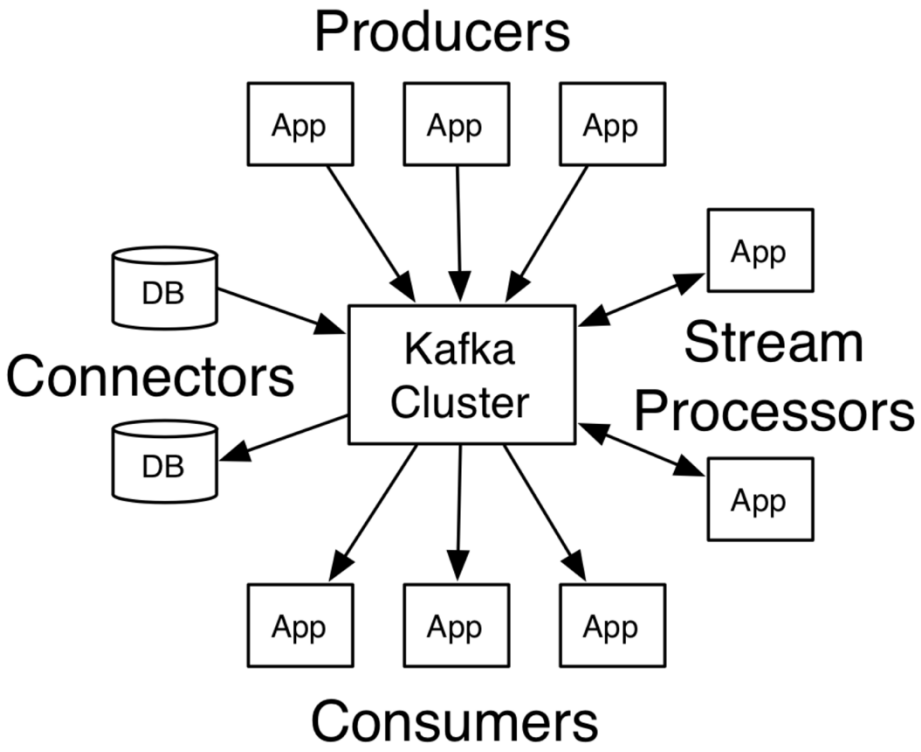
If you've ever built real-time data pipelines or streaming applications, you know how useful the Apache Kafka™ distributed streaming platform can be. Then again, you've also probably bumped up against the challenges of working with Kafka.

If you're new to Kafka, or ready to simplify your implementation, we present common challenges you may be facing and five ways that StreamSets can make your efforts much more efficient and reliable:

But First... A Short Kafka Primer

Apache Kafka is a distributed platform for running real-time streaming data pipelines and applications. Kafka allows you to:

- **Publish & subscribe.** Read and write streams of data like a messaging system.
- **Process.** Write scalable stream processing applications that react to events in real time.
- **Store.** Store streams of data safely in a distributed, replicated, fault-tolerant cluster.



Source: kafka.apache.org

Development teams are adopting Kafka as a means to implement a modern message bus, rethink how they perform extract, transform and load (ETL) operations, and support the shift to microservices architectures.

The business benefits from Kafka are many. Financial services companies use Kafka to help combat fraud, retailers use it to derive real-time customer insights, and manufacturers use it to improve product quality.

The Challenges of Working with Kafka

Kafka is fast, scalable, and durable. It's a really powerful tool for enabling event handling, and is extremely popular given its open source nature.

Still, you'll face certain challenges when working with Kafka. These challenges include:

- **Lots of custom coding.** Kafka requires specialized coding skills, including plenty of experience with Java and Python. Kafka developers are an expensive and scarce resource. *CIO Magazine* recently listed Kafka coding as one of the ten hardest-to-find tech skills, and salaries for positions on Indeed.com requiring Kafka skills range from $100,000 to more than $150,000. Scaling to dozens of developers can be both a recruiting challenge and a budget-breaker.
- **Fragmented connectivity.** Real-time streaming architectures have lots of moving pieces, and they come in a diverse range of configurations. Each data store you're using for streaming data, whether it's Hadoop, MongoDB or Oracle, requires a specific connector for Kafka. While many Kafka connectors exist, finding a single place to get support for all of them is hard. Any given Kafka vendor only supports a handful of these connectors, leaving you to custom code the rest.
- **Multiple stream processing frameworks.** Kafka is popular, but picking a stream processing framework to use with it is easier said than done. Kafka at it's core is a publish-subscribe message bus, Kafka Streams was added much later to provide stream processing support. But many stream processing frameworks came to market prior to Kafka Streams, including Apache Spark™ Streaming, Apache Flink™, and Apache Samza™. As the streaming market is still maturing, it's difficult to pick a long-term winner with confidence, and it's likely there will be different winners for different use cases. If you place your bet on coding solely to Kafka, you put yourself at risk of having to start all over again if you change or add other platforms.

- **Data drift and constant pipeline maintenance.** A key reason to implement Kafka is to simplify connecting all sorts of new data types to multiple processing techniques. But this new world is plagued by data drift – unpredictable and unending changes to data structure and semantics. Data drift makes data pipelines (including those connecting to Kafka) very high maintenance. At its core, Kafka and Kafka Connect are not designed to deal with the fact that data is frequently changing.

StreamSets – Taming Kafka through Data Operations

Dealing with continuous dataflows requires you to take an operational approach to the management of your Kafka pipelines. StreamSets has built a data operations platform to help you manage across the lifecycle of dataflow logic, not just for Kafka but across a wide variety of sources and stores.

The StreamSets platform helps you:

- **Build** dataflow pipelines in hours, not days or weeks, to increase the speed and productivity of your developers.
- **Execute** batch and streaming pipelines anywhere—on premises, in the cloud, or at the edge— so you can run end-to-end flows across any data architecture.
- **Operate** continuously with high reliability even as your data drifts and your architecture evolves.

Global 2000 companies across many industries use StreamSets for a variety of purposes, such as data lake ingestion, multi-cloud data movement, cybersecurity, IoT and customer 360 applications

StreamSets Advantage #1: Build Kafka Pipelines without Coding

StreamSets lets you build any-to-any batch and streaming pipelines quickly—all without custom coding. Integrating directly with Kafka, it helps your developers efficiently build, test and execute dataflow pipelines connecting myriad data sources through Kafka to multiple compute platforms such as Hadoop, NoSQL and search engines.

You can either build individual pipelines in StreamSets Data Collector, our open source software, or use StreamSets Control Hub to build and interconnect multiple pipelines into topologies, while allowing developers to collaborate and share best practices via a central pipeline repository. In either case you get a

drag-and-drop interface, an integrated development environment and numerous connectors and processors.

Of course there are times when only custom code will do, for instance for more complex tasks that are not easily handled by the built-in processors. For those cases StreamSets supports the ability to plug custom code into pipelines as needed.

What does that mean for you? Kafka pipelines built in hours not weeks, and development teams that are an order of magnitude faster and more efficient.

StreamSets Advantage #2: Insulate your Dataflows from Data Drift

A big challenge with hand-coded pipelines is that they often break when the data source changes or data platforms get updated. StreamSets is designed to provide resilient pipelines in two ways. First, it lets you quickly create Kafka consumers and producers by specifying only the fields you want to act on, something we call being “intent-driven”. So a change to a field doesn’t necessitate pipeline rework. Second, StreamSets has unique “pipeline sensors” which actively deal with data drift by monitoring the data, detecting and handling changes to schema and semantics. It can alert you to changes in the data stream as well as execute automated actions, such as immediately adding a new field to a downstream data store.

What does this mean for you? No more tying up developers with endless maintenance of brittle, hardwired and unreliable pipelines.

StreamSets Advantage #3: Effortlessly Blend Multiple Platforms

Kafka Connect is an example of a technology-specific data movement technology. As it turns out, nearly every data platform comes with its own specialized on-ramp technology, usually requiring hand coding and tied solely to that platform.

In contrast, you can think of StreamSets as your best-of-breed Switzerland of connectivity. It includes 100+ connectors and transformation processors that give you the flexibility to choose whatever data sources and systems you need, regardless of whether it’s Kafka, Spark, another streaming platform, or all of the above. Build whatever architecture suits your needs, and know that StreamSets can manage data movement across it.

What does that mean for you? You don't have to standardize on a single streaming platform, you don't need an army of developers to move data across platforms, and you are future-proofed against inevitable architecture change.

StreamSets Advantage #4: Scale On, but with Control!
Success means scaling, but also increased complexity as more and more teams build pipelines to leverage Kafka. As you scale you need to control deployment, manage pipeline versions and maintain end-to-end visibility into all of your data movement. As for data governance and problem diagnosis, you need to know where the data came from, where it's going, how it's being manipulated and who's interacting with it.

To simplify scaling for Kafka and all of your data movement, StreamSets Control Hub automates the build, deployment and management of your pipelines at scale. Its shared repository and version management features allow for collaborative development while also giving you control over what pipelines go into production. As your topologies evolve you get a complete map of your data movement, performance comparisons across versions and lineage information to help with governance and root cause analysis.

What does that mean for you? Control, even as your implementations become larger and more complex.

StreamSets Advantage #5: Operate Continuously Across Your Enterprise
Once you're in operation, the challenge becomes how to continually monitor data movement in and out of Kafka as well as across your broader architecture. You need to be able to answer questions like: Is your data moving fast enough? Are there any data quality issues you need to be aware of? Are you maintaining your SLAs to the downstream processes and applications that rely on timely and accurate data delivery?

StreamSets Dataflow Performance Manager (DPM™) answers these questions with fine-grained real-time metrics that offer point-in-time insight into how data is flowing across an architecture, plus visibility into the quality of that data. You can use those metrics to set and enforce Data SLAs. In short, it gives you complete operational visibility across your end-to-end data movement architecture.

What does that mean for you? Insight into how data is flowing at any point in time and an early warning system so you can manage your data architecture with confidence.

StreamSets + Apache Kafka = A Proven Combination

Remember those challenges of working with Kafka? We believe the StreamSets Data Operation Platform alleviates many of them, with a complete approach that covers the build, execute and operate steps in the lifecycle of data movement.

Enterprises in a wide range of industries are already benefiting from the combination of StreamSets and Kafka. Here are just a few examples:

- A leading healthcare company saved $1.4 million in pipeline development costs.
- A global financial services firm was able to deliver data to and through Kafka in 2 hours rather than 60 days.
- A market intelligence agency was able to staff an ingest project using 2 developers instead of 6, freeing valuable resources for higher value work.

Ready to Get Started?

Are you ready to make StreamSets part of your Kafka deployment?

Getting started is as easy as taking advantage of the free download of the open-source StreamSets Data Collector software.

To learn more about how StreamSets conquers dataflow chaos, visit the StreamSets website. To see how StreamSets works with Apache Kafka, check out this on-demand webinar: "Apache Kafka Made Dead Easy: Modern Data Ingestion Best Practices" or this short video "Create Kafka Pipelines in Minutes".