signal fx

KAFKA + SIGNALFX INTEGRATION

What is Kafka?

Apache Kafka is an open-source, distributed publish-subscribe message bus designed to be fast, scalable, and durable. Kafka provides a unified, high-throughput platform for handling real-time data feeds, maintained in topics, and can support a number of consumers and manage large amounts of data with very little overhead. It's used for log aggregation, message brokerage, activity tracking, operational metrics, and stream processing.

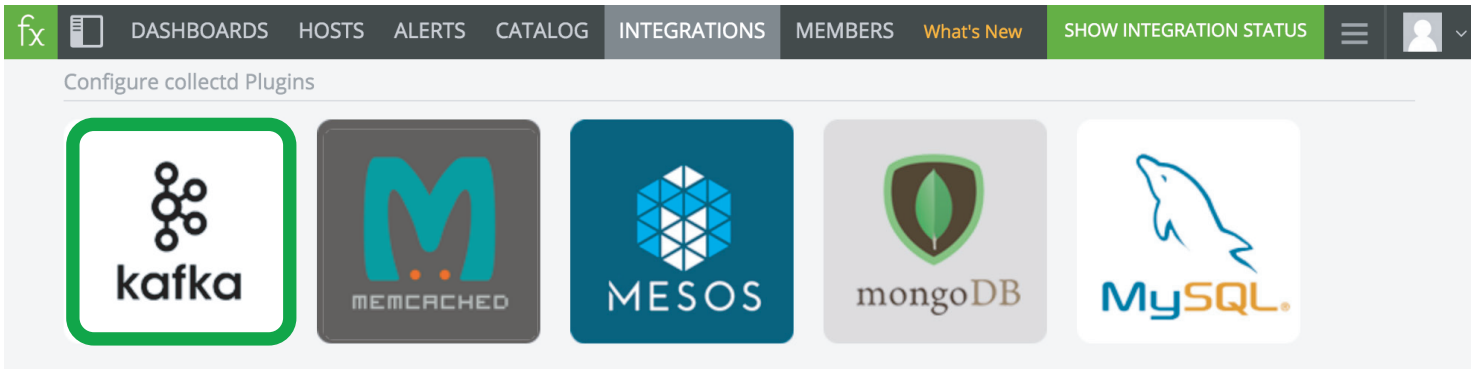
Sending Kafka Metrics to SignalFx

Use collectd with the GenericJMX collectd plugin configured to collect metrics exposed by Kafka via JMX, particularly for brokers and topics. In order to know what other services in the architecture are producing or consuming messages and of what size, wrap the client in an instrumented layer and collect system metrics for comparison against Kafka's metrics — this is an established pattern. SignalFx provides built-in dashboards displaying the most useful metrics for running Kafka in production and templates for adding topic names for topic-specific charts.

Monitoring Kafka

From SignalFx's experience monitoring Kafka in production, there are two primary leading indicators of issues: log flush latency and under-replicated partitions. In most cases, the changes are at the broker, not cluster, level. Although there's value to investigating bytes per message and approximate end-to-end latency, alerting on just the two leading indicators will result in meaningful notifications.

LOG FLUSH LATENCY: The longer it takes to flush log to disk, the more the pipeline backs up. A backed-up log pipeline is probably the most common cause of low throughput and high latency. For LogFlushRateAndTimeMs, monitor changes in the 95th percentile to understand outliers. If this number increases — even from 10ms to 20ms — end-to-end latency will often balloon and impact not just Kafka performance, but also performance of the entire system. Some topics are more latency-sensitive, so set alert conditions on a per-broker basis. Each broker's metrics have metadata that we apply (as key-value pairs of property:value) to identify the topics impacted. For example, raw customer data being ingested is highly latency-sensitive, so it gets a 15ms threshold.



UNDER-REPLICATED PARTITIONS: This indicates replication is not going as fast as configured, which adds latency as consumers don't get the data they need until messages are replicated. It also suggests more vulnerability to losing data if there is a master failure. Any under-replicated partitions that last more than a minute constitute a bad thing. For alerting on UnderReplicatedPartitions, use a simple greater-than-zero threshold against the metric exposed from Kafka and a duration condition.

Kafka Metrics

Bytes In	Bytes Out
Active Controllers	Request Queue
Log Flushes	Log Flush Time in Ms
Produce Total Time	Produce Total Time – 99th Percentile
Fetch Consumer Total Time	Fetch Consumer Total Time – 99th Percentile
Fetch Follower Total Time	Fetch Follower Total Time – 99th Percentile
Messages In	Produce Total Time – Median
Under Replicated Partitions	Fetch Consumer Total Time – Median
Log Flush Time in Ms – 95th Percentile	Fetch Follower Total Time – Median

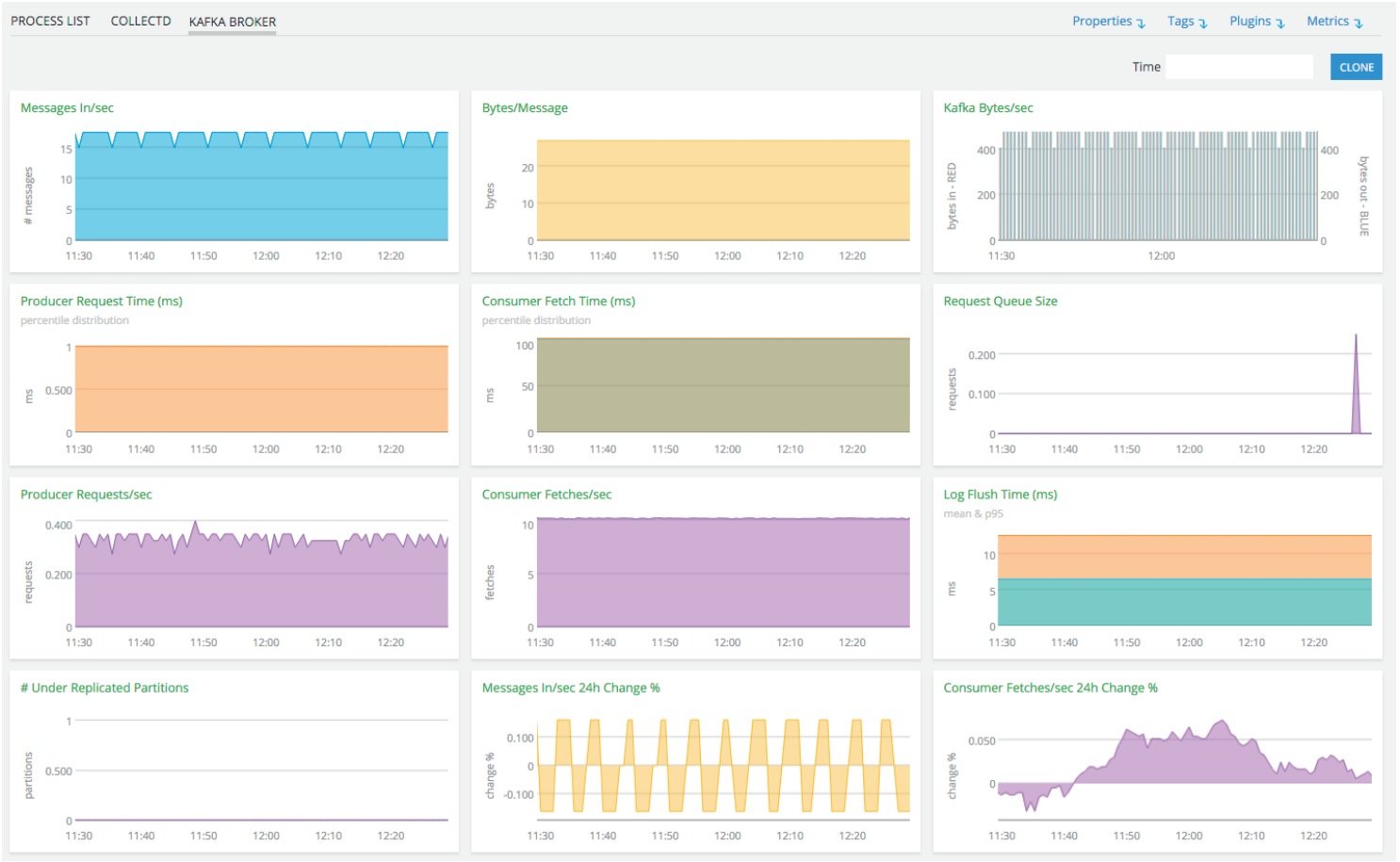
BYTES PER MESSAGE: Messages and bytes in represent how well-balanced traffic is. A change in their standard deviations indicate a broker is (or multiple brokers are) overloaded. Using MessagesInPerSec (per broker or per topic) and BytesInPerSec (per broker or per topic), you can derive the combined metric of bytes per message. An increase in this metric should be a cause for concern, unless the developers intentionally wanted large messages — by default, Kafka works best with smaller messages.

The SignalFx Difference

CREATING DERIVED METRICS: Individual metrics sometimes don't provide enough insight on their own to serve as the basis of meaningful, reliable alerts. As with any complex service operating at large scale, understanding Kafka's performance relies on dependencies, not just among other services in the architecture, but also among the various Kafka consumer metrics themselves. Specifically, seeing the standard deviations of bytes per second or messages per second alone is often not enough information to identify a problem. Deriving bytes per message, however, is a more meaningful way to evaluate a topic's or broker's behavior. Another common derived metric is read amplification, derived from bytes in per second and bytes out per second. SignalFx makes it easy to visualize custom metrics in real time without writing code or having to wait to perform the calculations after the fact.

SCALING WITHOUT MESSAGE LOSS: Scaling Kafka can get complicated when brokers are heterogeneous. The topic partitions have to get spread across new brokers if there is heavy traffic on one topic. Migrating many partitions, each with a lot of data, can easily saturate the network and increase the likelihood of message loss. SignalFx helps reduce migration time and risk by monitoring network in and out from collected on the source and target brokers. Use those metrics to control the pace of rebalancing and prevent resource starvation so app-level performance is not impacted by service-level changes.

CURATING METRICS AND GETTING VISIBILITY: There are many metrics specific to Kafka, and knowing where to start and what to monitor can be difficult. Similarly, Kafka doesn't operate in isolation, so analyzing a single service's data without the context of other application and system metrics from across the infrastructure restricts value. SignalFx curates the Kafka metrics that matter right out of the box based on experience with large, diverse architectures in production. SignalFx also provides pre-built dashboards, meaningful alerts, and automatic configuration that give you a running start on monitoring your modern environment.



About SignalFx

SignalFx is the most advanced monitoring solution for modern infrastructure. Our mission is to help cloud-ready organizations drive high levels of availability in today's elastic, agile, distributed environments. With SignalFx, development and operations teams gain a real-time view of, interact with, and take action on the infrastructure and application metrics that matter. We have enterprise customers including Yelp, Cisco, Zuora, Hubspot, and thousands of users analyzing billions of metrics every day. SignalFx was founded in 2013 by former Facebook and VMware executives, launched in 2015, and is backed by Andreessen Horowitz and Charles River Ventures.