



WHITE PAPER

# The Secrets To Successfully Monitoring Fast Data And Streaming Applications



Lightbend

# Table of Contents

<b>Executive Summary.....</b>	<b>2</b>
<b>Key Takeaways .....</b>	<b>2</b>
<b>Introduction.....</b>	<b>5</b>
Big Data In Batches vs. Fast Data In Streams .....	6
<b>The Challenges Of Monitoring Fast Data Applications .....</b>	<b>7</b>
Rapidly Evolving Ecosystem .....	8
Understanding The Data Pipeline .....	8
Dynamic Architectures.....	8
Intricately Interconnected .....	8
Distributed And Clustered.....	9
Apache Spark, As An Illustrative Example .....	10
<b>What To Look For In A Fast Data Monitoring Solution .....</b>	<b>11</b>
Traditional Monitoring Tools And Fast Data Applications.....	11
APM And Infrastructure Monitoring Tools.....	12
Five Key Capabilities To Ensure Application Health .....	12
<b>Intelligent, End-To-End Monitoring From Lightbend .....</b>	<b>12</b>
Intelligent, Data Science-Driven Anomaly Detection .....	13
Automated Discovery, Configuration And Topology Visualization.....	14
Intelligent, Rapid Troubleshooting.....	15
<b>The Business Value Of Lightbend Monitoring For Fast Data Applications.....</b>	<b>16</b>
Increase Customer Satisfaction.....	17
Reduce Costs .....	17
Realize Value Quickly.....	17
<b>Conclusion .....</b>	<b>17</b>

# Executive Summary

The increasingly real-time requirements of today's applications are changing how users expect services and products to be delivered and consumed.

Enterprises are responding to this by embracing Reactive system architectures coupled with best-in-class data processing tools to create a new category of programs called Fast Data applications. These applications are sparking the emergence of new business models and new services that take advantage of real-time insights to drive user retention, growth, and profitability.

While Fast Data applications are powerful and create significant competitive advantages, they also impose challenges for monitoring and managing the health of the overall system. Traditional monitoring solutions, built for legacy monolithic applications, are unable to effectively manage these intricately interconnected, distributed, and clustered systems. Businesses must therefore rethink their approach if they wish to take full advantage of the Fast Data revolution.

This white paper outlines the functions a modern monitoring solution must perform in order to truly benefit from the advantages promised by Fast Data and streaming applications.

## Key Takeaways

- Reactive system architectures are ideal for building, deploying, and managing Fast Data applications, which deliver a significant competitive advantage by enabling enterprises to identify and seize opportunities faster.
- These architectures bring services closer to the data stores by building data streams into the applications, enabling real-time personalization, real-time decision-making, and IoT data processing, and providing the opportunity to modernize legacy batch processing systems.
- New open source technologies such as Apache Spark, Apache Mesos, Akka, Apache Cassandra, and Apache Kafka (a.k.a. the “SMACK” stack) provide a complete set of components to rapidly build powerful Fast Data applications.
- To effectively monitor these applications, Enterprises are challenged with monitoring and troubleshooting constant streams of data from dozens or even hundreds of individual, distributed microservices, data sources, and external endpoints.
- Current monitoring tools, designed for simple monolithic systems, don't work well for these Fast Data applications.

- Lightbend Monitoring provides deep telemetry to gain visibility into the right operational metrics for applications, real-time understanding of application health and data pipelines, and a powerful visualization layer that shows the end-to-end health, availability/performance of apps, data frameworks, and infrastructure in a single view.

# Introduction

 Businesses need to be Reactive because you can't predict the future. They will need new technical architectures to support the change, which looks a lot more like web computing: agile, bursty, lean. That's the future of business.

—James Governor, RedMonk

Over the last two decades, and especially in the last few years, the computing infrastructure landscape has dramatically changed. Cheap multi-core processors are now ubiquitous. Clusters of servers, powered by ever more powerful processors, are commonplace. Disk storage has become a commodity. Mobile devices in every form factor have proliferated. Networks have improved speeds significantly and now connect users throughout the world anywhere and anytime.

Business of all types, from nimble startups to established enterprises, can now build new applications or innovate on existing ones, in ways that take advantage of this changed computing landscape. But if the capabilities of applications from competitors and disruptors are increasingly similar, what differentiates one application from another is the user experience. This is of vital importance because the user experience increasingly correlates with retention, growth and profitability.

Legacy	Modern
Single machines	Clusters of machines
Single core processors	Multicore processors
Expensive RAM	Cheap RAM
Expensive disk	Cheap disk
Slow networks	Fast networks
Few concurrent users	Lots of concurrent users
Small data sets	Large data sets
Latency in seconds	Latency in milliseconds

However, user expectations and needs are constantly evolving. So while enterprises need to ensure that users have a highly responsive experience at all times with their applications, they also need to have the ability to continually roll out new features and capabilities. This, in turn, is driving a monumental shift in the industry to the new paradigm of Reactive systems.

Reactive systems are designed to maintain a level of responsiveness at all times, elastically scaling to meet fluctuations in demand and remaining highly resilient against failures. The three tenets of Reactive systems, along with the message-driven design philosophy that makes these tenets possible, were first codified in the [Reactive Manifesto](#) in 2013 by Lightbend.

Since then, Reactive has gone from being a virtually unacknowledged technique for constructing applications—used by only fringe projects within a few corporations—to becoming part of the overall platform strategy for numerous big players in the middleware field. Fueling this trend, in addition to responsiveness, is the fact that enterprises that build Reactive systems experience a significant boost in developer productivity and corresponding increase in release velocity.

At the same time, there has been an explosion in the volume of data generated and collected by businesses everywhere. Technologies such as Apache Spark, Apache Kafka, and Apache Cassandra have arisen to process that data faster and more effectively.

But data that is merely collected and analyzed after the fact is of limited use to businesses. What if, instead, this data could be analyzed in real-time and generate insights at the touch of a button? What if businesses could learn from their users' historical data and be primed to make real-time decisions that serve clients better? What if businesses could analyze data from sensors and devices in real-time to automatically optimize and tune the underlying infrastructure, driving massive cost efficiencies?

To answer those questions, businesses across a variety of verticals are pushing a new wave of application innovation, combining Reactive systems with data processing tools. These new applications are being referred to as *Fast Data applications*<sup>1</sup>.

## Big Data In Batches vs. Fast Data In Streams

Compared to the batch-mode, “data at rest” practices of traditional Big Data systems, the ability of Fast Data applications to process and extract value from data in near-real-time has quickly become a key differentiator for modern businesses.

Even for data that doesn’t strictly require real-time analysis, the importance of streaming has grown in recent years because it provides a competitive advantage that reduces the time gap between data arrival and information analysis.

---

<sup>1</sup> Read more in [Fast Data Architectures For Streaming Applications \(O'Reilly\)](#), by Dean Wampler

Reactive systems and the Fast Data applications running on them offer many strategic and operational advantages. By adapting quickly to evolving data sets, organizations can stay on top of market trends and maintain a live, relevant, experiential relationship with their customers. Like the code version of the Transformers® toys, Fast Data applications can automatically reshape themselves to serve new requirements in a rapid cycle.

Several B2B industries, including financial services, retailers, marketing, security, and healthcare, are leveraging real-time (or near-real-time) insights to enable transformative business impact for their customers through:

- Real-time personalization
- Real-time decision-making
- IoT data processing
- Legacy batch processing modernization

By embracing Fast Data applications and taking advantage of previously undetectable data-driven insights, businesses are seeking to retain existing customers through superior levels of service, attract new ones through innovative business models, and enter new markets and segments rapidly.

But adopting this new approach to data streaming is only one half of the equation. The main challenge is how to assure the continuous health, availability, and performance of these modern, distributed Fast Data applications. And that isn't always easy.

## The Challenges Of Monitoring Fast Data Applications

Fast Data applications constantly stream data from dozens or even hundreds of individual, distributed microservices, data sources, and external endpoints. Thanks to the growing popularity of new open-source technologies such as Apache Spark, Apache Mesos, Akka, Apache Cassandra, and Apache Kafka (a.k.a. the “SMACK” stack), businesses can now utilize a complete set of components to rapidly build powerful data processing applications.

However, the distributed and complex nature of these new applications makes monitoring them quite challenging, for a number of reasons:

## Rapidly Evolving Ecosystem

As numerous data processing frameworks continue to appear on the horizon in a relatively short time, domain knowledge is relatively scarce. Identifying which metrics to collect and understanding the business value behind them is by no means an easy undertaking. The learning curve can easily take months.

## Understanding The Data Pipeline

Enterprises adopting Fast Data applications and streaming data need to consider the data pipeline as a core component. A data pipeline is a set of explicitly-defined data processing elements connected in series, where the output of one element is the input of the next.

- Throughput
- Error Rate
- Latency
- Backpressure
- Data Loss

Things can go wrong in any stage, making it imperative to compute those metrics both on a per-stage basis, *and* across the entire defined pipeline, for a holistic understanding of application health and performance.

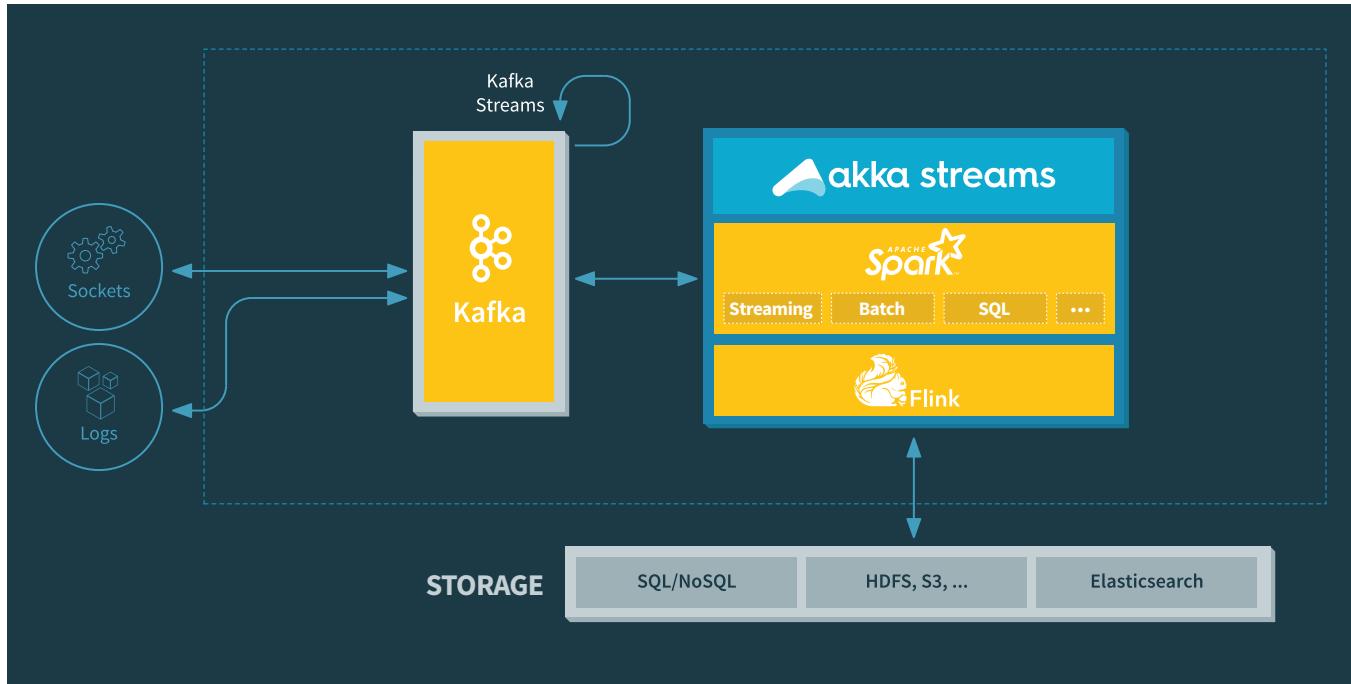
## Dynamic Architectures

Application infrastructures are no longer static. Instead, they change and grow over time in response to workloads, requirements, or new business services. This means that it's impractical to manually retrieve relevant data, configure and calculate the desired aggregations, create the desired dashboards, and set up the appropriate monitors. Automation wherever possible is essential to saving time and decreasing the risk of manual errors.

## Intricately Interconnected

A Fast Data application is a complex system. The figure below shows one such application, that's comprised of many parts - data routing, data processing, storage, resource allocation, and recovery. Host

management and job management, though not shown below, are also part of the architecture. Applications like this are typically deployed on clusters of machines, which may serve different functions and have dependencies on other services or infrastructure components.



Technologies such as Akka, Spark, Kafka, Mesos and Flink all fall under this category. Problems that manifest in one part of the system can often originate in a completely different place. For example: imagine a Spark job that is reporting a drop in data processing throughput. Where should IT start to look? Is the problem upstream in Kafka or within the system that is feeding Kafka? Is it a downstream problem with ElasticSearch in writing data to the store? Is there something wrong with the Spark job itself? In these increasingly distributed and complex systems, it can be difficult to know where to start looking when a problem emerges.

## Distributed And Clustered

Each framework is composed of several components, usually deployed in a distributed environment. Apache Spark, for instance, consists of master, worker, application, driver, and executor. The complexity of the data pipeline grows exponentially as multiple data frameworks are stacked up together and intertwined with custom application code across distributed clusters. Logs, the traditional mechanism for identifying and tracing issues, can be hard to read or query, lack user context, be spread across multiple servers, or be void of helpful details.

Successfully monitoring these applications requires collecting metrics and performing checks on several data frameworks, custom code, and dozens or hundreds of hosts. Correlating issues to understand dependencies and analyze root causes is therefore critical.

## Apache Spark, As An Illustrative Example

With each component generating its own metrics and data that may be dependent on other components, it's easy for engineering to get overwhelmed by the flood of incoming data. The first step in trying to solve this problem is to properly organize information into a hierarchy of concerns.

Here is an illustrative example that hones in on just Apache Spark, though this approach can be generalized for other components of Fast Data applications as well:

Concern	Questions To Ask
Data Health (for a given application)	<ul style="list-style-type: none"><li>• Throughput: is data processing occurring at the expected rate?</li><li>• Latency: is data processing occurring within the expected timeframe?</li><li>• Error/quality: are there problems with the data being produced?</li><li>• Input data: are input data streams flowing into Spark behaving normally? For instance, what are the throughput rates for Kafka topics feeding into the Spark job?</li></ul>
Dependency Health	<ul style="list-style-type: none"><li>• Are the systems feeding input into the storm job (such as Kafka) healthy?</li><li>• Are the systems that the application is dependent on, such as Memcached or other API endpoints, healthy?</li></ul>
Service Health	<ul style="list-style-type: none"><li>• Is the Spark master operating normally? If not, engineering will be unable to re-balance workloads or restart jobs.</li></ul>
Application Health	<ul style="list-style-type: none"><li>• Are the application KPIs within normal operating parameters?</li></ul>
Topology Health	<ul style="list-style-type: none"><li>• Are there resources assigned to the given Spark topology?</li><li>• Are the Spark tasks and executors well-distributed amongst the Spark cluster?</li><li>• Are the performance counters (emitted, failed, latency, etc.) for the given Spark topology normal?</li></ul>
Node System Health	<ul style="list-style-type: none"><li>• Are the key system metrics (load, CPU, memory, net-i/o, disk-i/o, disk free) operating normally?</li></ul>

Ideally, all data relevant to each concern area should be available and visualized in one place, enabling faster situational understanding. But that is generally not the case. And when expanding the scope from just Spark to other components, the size of the challenge becomes even more daunting.

The question then is whether traditional monitoring solutions can effectively rise up to the challenges of monitoring Fast Data applications and their components.

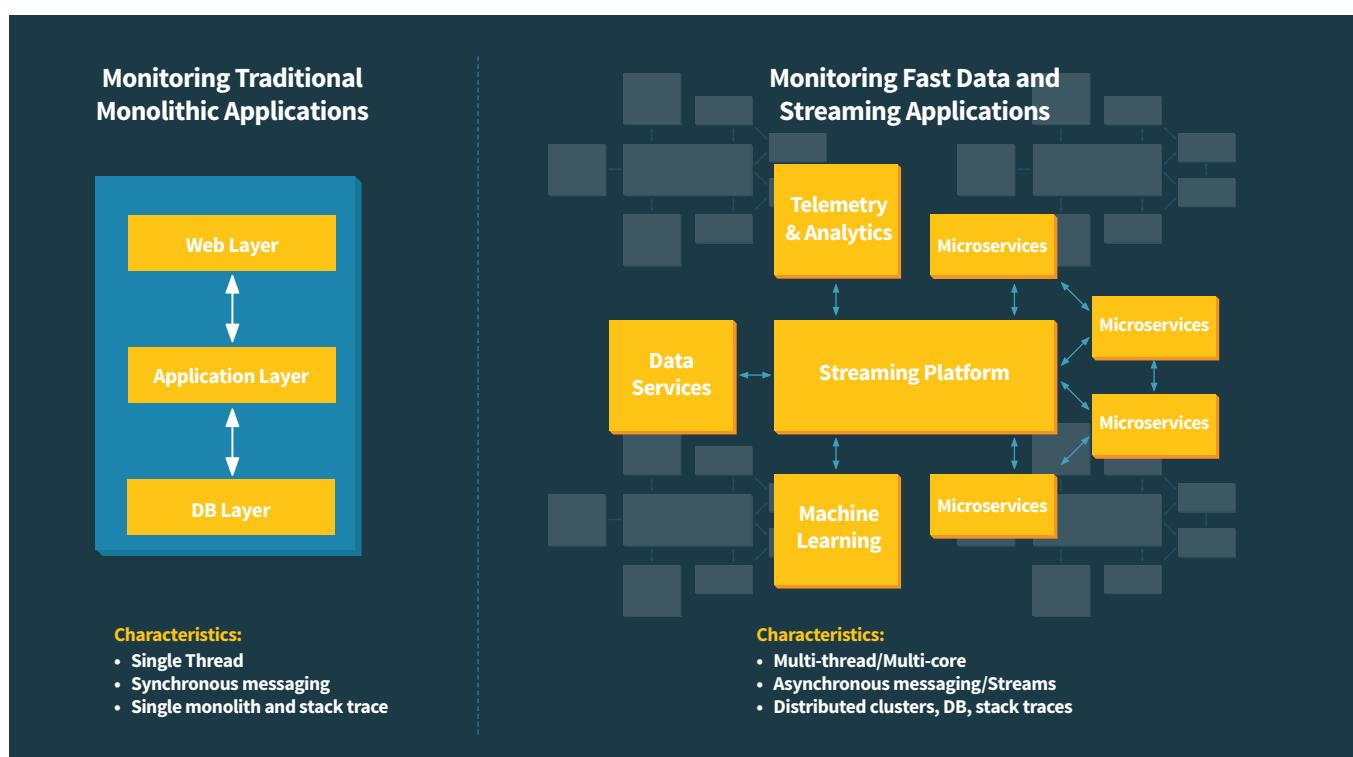
## What To Look For In A Fast Data Monitoring Solution

### Traditional Monitoring Tools And Fast Data Applications

For the most part, current monitoring philosophy for software is based on decades-old monolithic design, where the database layer, application layer, and front-end web layer are all packaged in a single “box”.

When an error occurs, the request causing the error runs on a single thread, resulting in a clear call stack from beginning to end. The call stack allows engineers to peek back in time to find the cause of the error. The key here is that the programmatic flow is deterministic, giving IT all the information required for debugging. As a result, users can extract metrics and trace information based on a synchronous flow.

To contrast, Fast Data applications are ideally asynchronous throughout and comprised of a combination of application code, microservices, data frameworks, telemetry, machine learning, the streaming platform, and an underlying, increasingly containerized infrastructure. The application code is in many cases also closely intertwined with the foundational data frameworks, making traditional monolith-oriented monitoring tools ineffective at monitoring Fast Data applications.



## APM And Infrastructure Monitoring Tools

Some Application Performance Monitoring (APM) tools have gained significant traction in the marketplace in the last few years, and were once viewed as a promising way to monitor Fast Data applications. Unfortunately, since these tools weren't designed from the ground up to monitor asynchronous, streaming systems running on distributed clusters, enterprises soon found them to be ineffective.

Similarly, infrastructure monitoring tools adapted for specific pieces of modern infrastructures have become available over the last few years. But again, these solutions fail to adequately monitor Fast Data components due to architectural interdependencies and component complexities.

Due to their highly purpose-built nature, log analysis and network performance monitoring tools are also incapable of successfully managing Fast Data applications.

## Five Key Capabilities To Ensure Application Health

To successfully achieve the full benefits promised by Fast Data and streaming applications, monitoring solutions should offer users the following:

- **Deep Telemetry:** Provide the right instrumentation to gain deep visibility into the real-time status of applications—starting in development for better testing, and flowing into production for live system insights
- **Domain Expertise:** Offer deep domain expertise on application components so that the solution automatically knows which metrics to monitor, with the flexibility to add custom metrics
- **Automated Discovery:** Automate the discovery of components that need monitoring as well as the setup of the monitoring environment itself
- **Real-Time Topology Visualization:** Help users visualize and understand the health of the application infrastructure and all associated components in real-time, so users can understand and track the health, availability, and performance of the entire application
- **Intelligent, Rapid Troubleshooting:** Allow solutions to learn from data over time, separating the signal from the noise and drilling down to root causes and problems quickly, thus optimizing uptime and lowering mean-time-to-repair

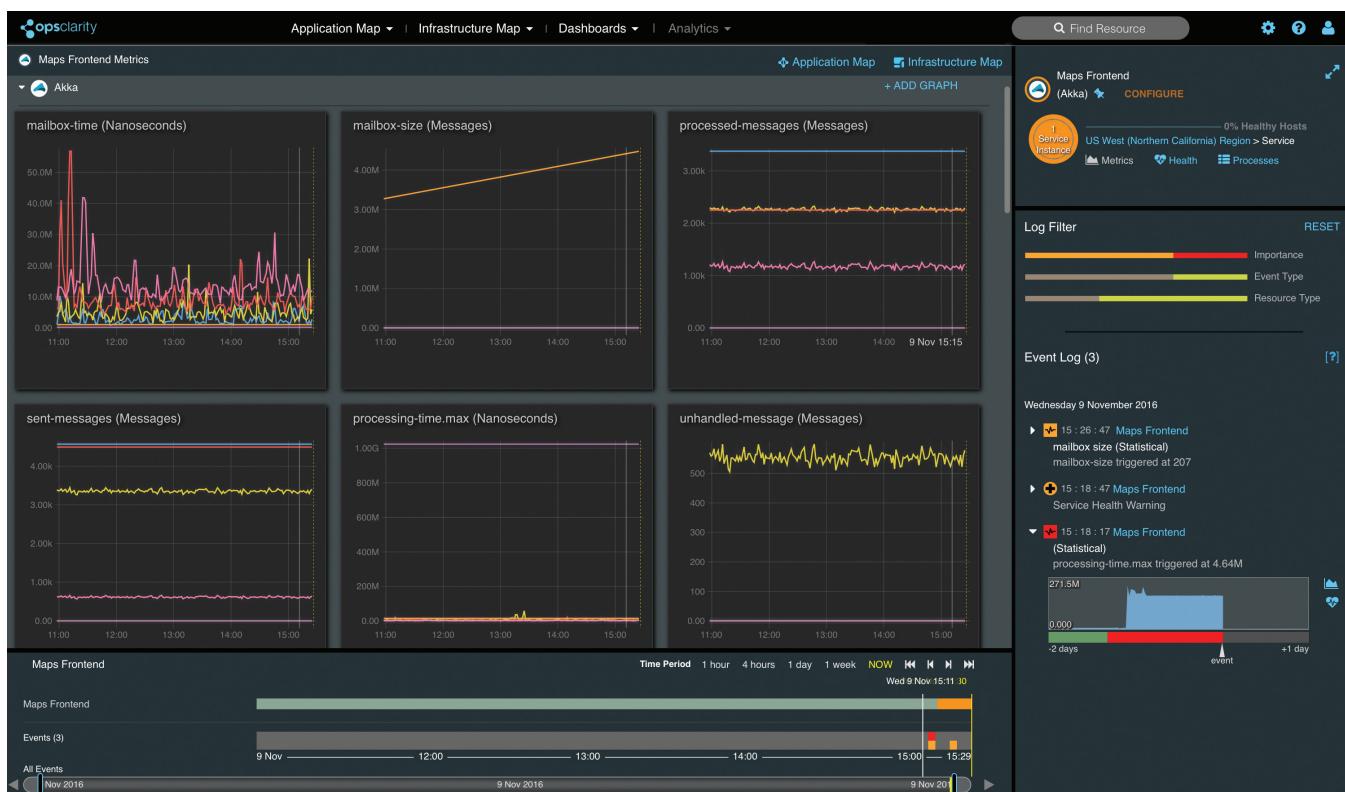
The next section examines Lightbend Monitoring, a modern approach to monitoring Fast Data applications that provides these key capabilities.

# Intelligent, End-To-End Monitoring From Lightbend

Lightbend Monitoring is a purpose-built monitoring solution for Fast Data applications that takes a modern approach to instrumenting and visualizing distributed streaming systems. Unlike traditional vendors, Lightbend Monitoring focuses on better controlling and maximizing the value of insights gathered from Fast Data applications in both development and production.

Lightbend Monitoring provides deep instrumentation to gain visibility into the right operational metrics for applications, real-time understanding of the health of applications and data pipelines, and a powerful visualization layer that shows the end-to-end health, availability, and performance of apps, data frameworks, and infrastructure in a single view.

## Intelligent, Data Science-Driven Anomaly Detection

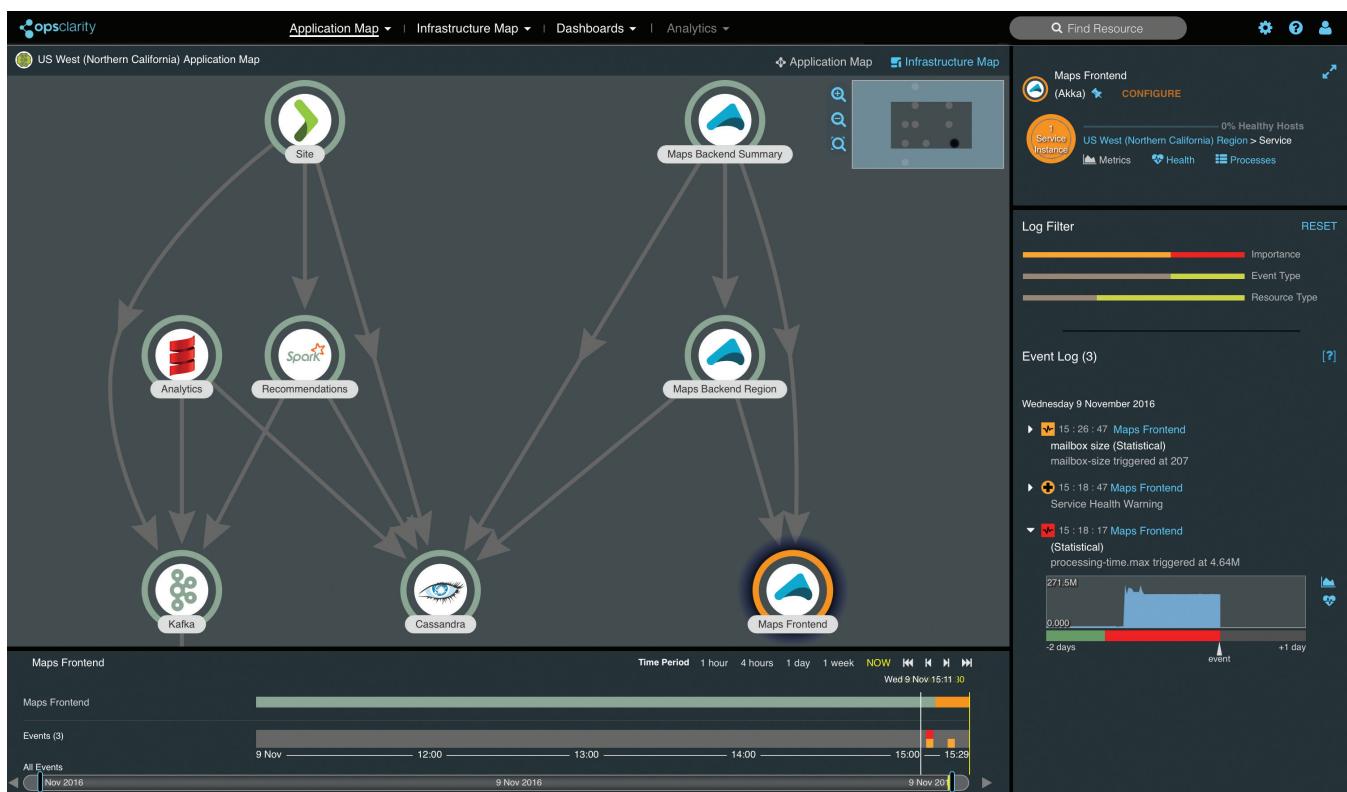


Lightbend Monitoring learns how to monitor a Fast Data application better over time, separating the signal from the noise of application processes and arms teams with the metrics they need.

- **Deep Telemetry.** Lightbend Monitoring consolidates multiple signals from across an entire Fast Data application into a single picture of the data pipeline and overall application health.
- **Domain Expertise.** Lightbend Monitoring automatically configures Fast Data applications based on understanding Fast Data components. For example, in Akka, this includes Metrics, Actor Events,

Threshold Events, and Trace Spans. For Spark, this covers various metrics for Spark Driver, Executor, Master, Worker, HDFS, and YARN. For Kafka, metrics are provided for Kafka Brokers, Producers, and Consumers.

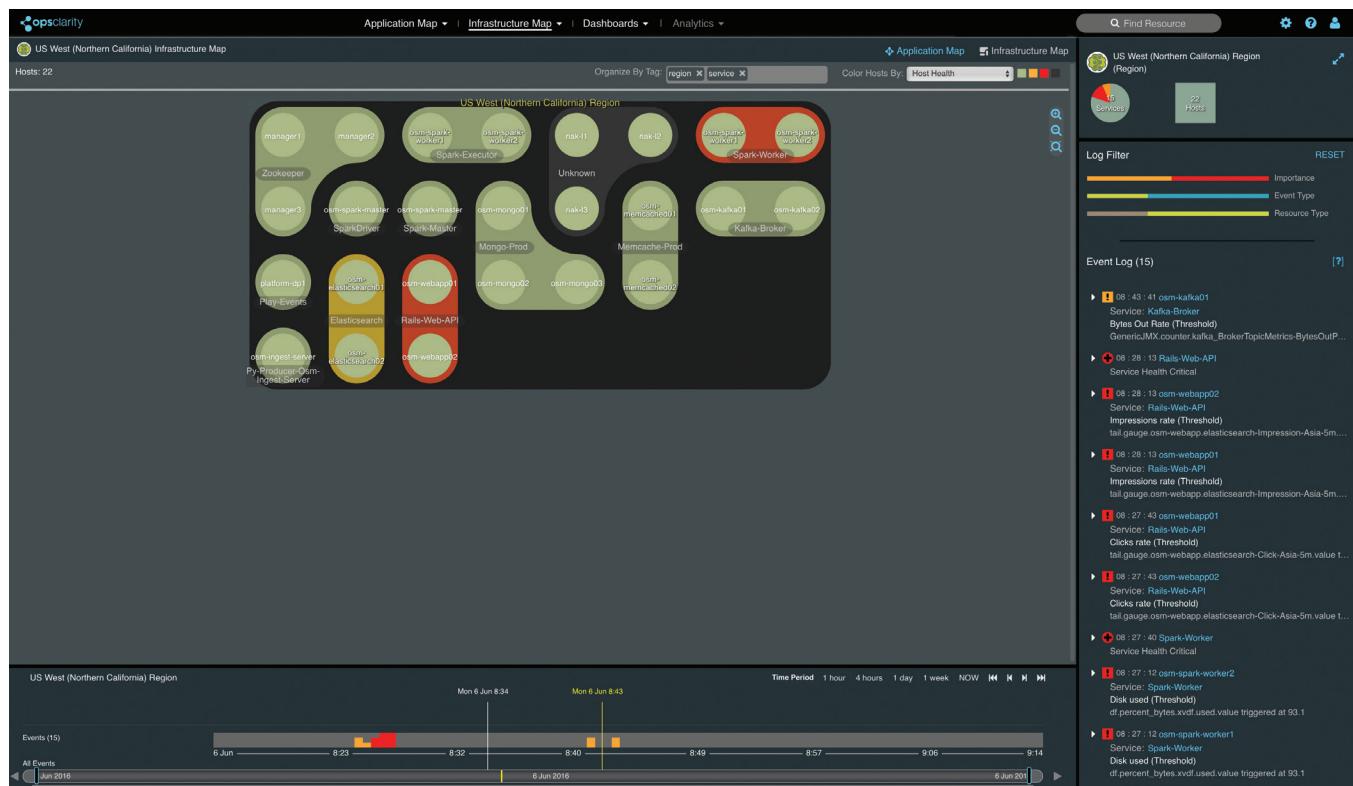
- **Intelligent Anomaly Detection.** Using built-in anomaly detection capabilities, Lightbend Monitoring intelligently learns which areas of the system experience which issues. This helps IT catch performance issues early on, before users are affected.
- **Fine-Grained Visibility, with Drill-Down Capabilities.** Lightbend Monitoring lets users drill down into the health of every service and infrastructure component in the Fast Data application, improving focus and minimizing data overload.



# Automated Discovery, Configuration And Topology Visualization

Lightbend Monitoring combines automated topology discovery and real-time visualization of the entire data pipeline. This enables early identification and quick resolution, even in development, of common concerns such as throughput, latency, error rate, and back pressure. This allows enterprises to:

- **Automated Topology Discovery.** Lightbend Monitoring provides auto-discovery of the complete topology of the Fast Data application, including operating systems, the network, and data.
- **Automatic Metric Collection.** Lightbend Monitoring eliminates the need for time-consuming manual metric and metadata collection, providing automatic, out-of-the-box metric configuration for a growing list of applications and frameworks.
- **Real-Time Topology Visualization.** Lightbend Monitoring helps visualize the status of different components of Fast Data applications in real-time, allowing users to easily track system health, availability, and performance metrics.

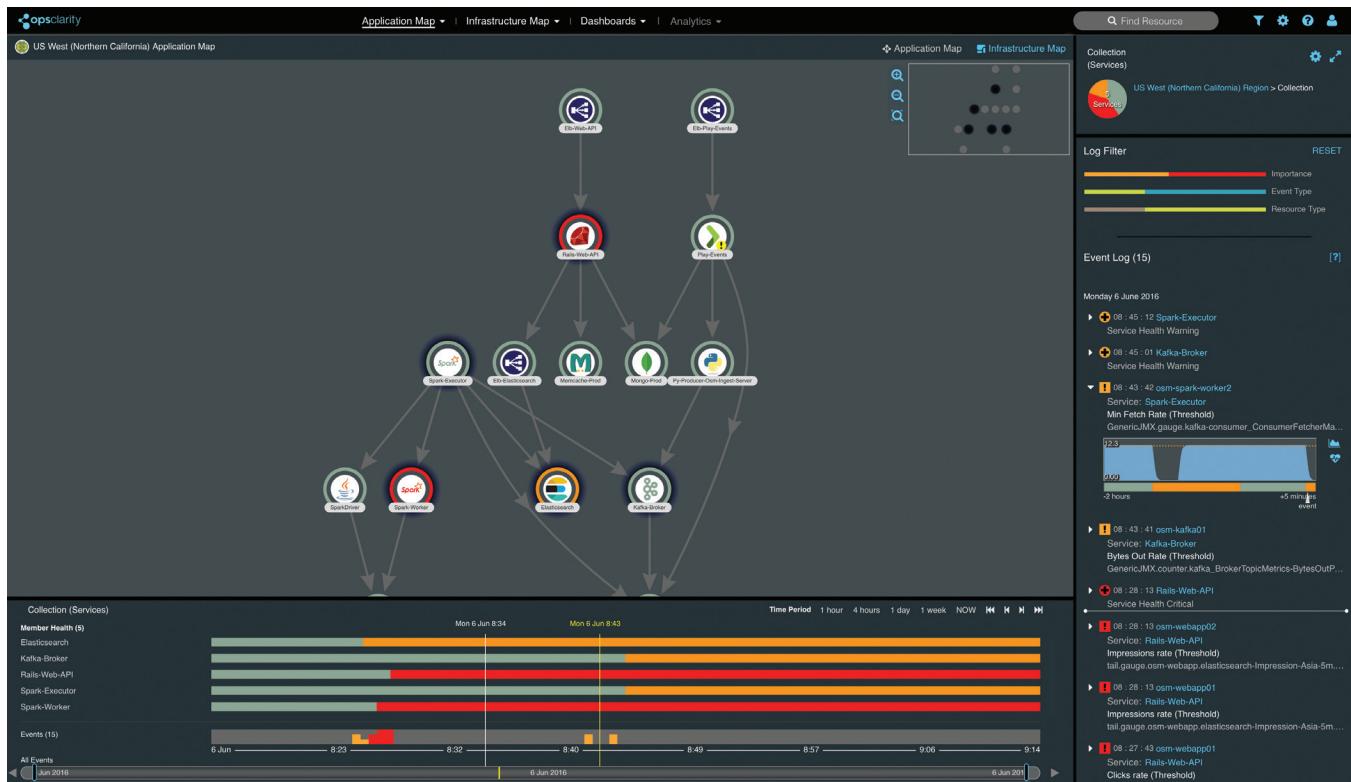


## Intelligent, Rapid Troubleshooting

Lightbend Monitoring provides users with the power to correlate events across their Fast Data applications, so they can gain the following capabilities:

- **Single Pane of Glass Visibility.** Administrators and managers have a holistic view of their system at their fingertips. Proactive alerts and dashboards keep everyone informed about system performance, while the local capture and aggregation of various operational and health metrics minimizes overhead and cost to applications.
- **Rapid Root Cause Analysis.** Getting to the heart of problems fast is invaluable. Lightbend Monitoring provides visibility across multiple service clusters and event replay for root cause analysis, enabling fast troubleshooting.
- **Reduced MTTR.** By detecting and isolating issues and problems, and helping troubleshoot them quickly, Lightbend Monitoring minimizes downtime, reduces the mean-time-to-repair (MTTR), and increases SLA adherence, leading to a superior level of service for users.

While these features are useful and benefit both production/ops and development teams, Lightbend Monitoring also delivers significant business benefits to enterprises.



# The Business Value Of Lightbend Monitoring For Fast Data Applications



*“Mobile and IoT use cases are driving enterprises to modernize how they process large volumes of data. Lightbend provides the fundamental building blocks for developing, deploying, and managing today’s large-scale, distributed applications.”*

—**Doug Fisher, Intel**

Lightbend Monitoring benefits business in three ways:

## Increase Customer Satisfaction

In today's highly-competitive, fast-moving environment, customers that experience outages or performance degradations are likely to leave for a competitor. Lightbend Monitoring not only helps eliminate outages and minimize downtime, but also helps catch hard-to-detect performance issues early on, increasing customer satisfaction and reducing churn.

## Reduce Costs

By helping minimize downtime, outages, and performance degradations, Lightbend Monitoring helps businesses avoid overspending on their hardware/infrastructure and avert downtime-related costs such as chargebacks and SLA penalties. This is in addition to any indirect costs to the brand reputation stemming from unexpected or prolonged outages.

## Realize Value Quickly

By packaging deep telemetry, automated discovery, configuration, topology visualization, and data-science-driven anomaly detection together into an easy-to-use solution, businesses can see and experience the value of Lightbend Monitoring provides within just days.

# Conclusion

In today's rapidly evolving, real-time world, businesses that take full advantage of Fast Data and streaming applications will win the day. This requires a new approach to architecting applications, one that is Reactive, streaming, and intelligent. But with that new approach come a number of monitoring and management challenges - issues that traditional, monolithic monitoring practices are woefully ill-equipped to resolve. Lightbend Monitoring provides intelligent, proactive visibility into the health, availability and performance of these Fast Data and streaming applications. This allows organizations to reliably serve their users with these applications, helping them thrive in the new real-time paradigm and surge past their competition.

**Contact us** to schedule your 20-minute introductory call and demo of Lightbend Monitoring, and see how your business can get ready for the Fast Data revolution.

# End-To-End Monitoring For Your Fast Data And Streaming Applications From Lightbend

SCHEDULE YOUR 20-MIN DEMO



Lightbend

Lightbend (Twitter: [@Lightbend](#)) provides the leading Reactive application development platform for building distributed applications and modernizing aging infrastructures. Using microservices and fast data on a message-driven runtime, enterprise applications scale effortlessly on multi-core and cloud computing architectures. Many of the most admired brands around the globe are transforming their businesses with our platform, engaging billions of users every day through software that is changing the world.