



signal fx

# WHY YOUR APM AND LOGS STILL NEED INFRASTRUCTURE MONITORING

A Modern Approach to Application Lifecycle Management

[www.signalfx.com](http://www.signalfx.com)  
[@signalfx](https://twitter.com/signalfx)

## The Problem with Pull-Based Monitoring

*It shouldn't be all about the consumer*



## The Old Way: Element Managers & Health Checks

Prior to the rise of the cloud, infrastructure health was primarily understood through the spectrum of simplistic IT health checks. Tools like Nagios and HP OpenView would pull status updates from the various machines and devices across the network. They'd report when any server or switch failed to respond to a ping or behaved out of the ordinary, and the infrastructure team would respond accordingly.

*Legacy monitoring tools arbitrarily pulled health checks across the network, prioritizing hardware failures.*

As the hardware and software stack became more complex, operations teams relied on additional information to build a more complete view of the state of their IT. In addition to health status, specialized application testing, network management, and server monitoring tools helped with performance engineering and analysis at each layer of the stack.

While a collection of element managers could provide specific insight into events at the database or storage layer, for example, so-called Manager-of-Managers technologies like IBM Tivoli, BMC PATROL, and CA Unicenter became necessary to capture, correlate, and make sense of the abundance of operations data.

Operations teams used a Manager-of-Managers to determine when a problem was significant enough to page someone in the middle of the night. However, as infrastructure and applications shifted to elastic, distributed cloud environments, traditional element and systems managers began to fail under the increased variety of data and complexity of performance requirements. While pinpointing the location of a down server was the largest priority for the infrastructure team under the old regime, the ephemeral nature of modern infrastructure requires a more service-wide view of availability.

# Gartner®

*“Got Nagios? Get rid of it. The problem is that the level of usability and sophistication of the product is pretty much zero.*

*The underpinning and ugliness still exist once you get through the layers intended to cover up the mess that Nagios is. There are still scripted ‘checks’ which run to determine service health, the checks are normally challenging to manage, especially when some execute through the agent, while others do not.*

*Nagios instances don’t auto-configure themselves, they don’t detect application instances properly or consistently, and configuration of checks is painful.*

*[I routinely tell clients] to ditch Nagios and go with a [better] monitoring tool.”*

**Jonah Kowall, “Got Nagios? Get Rid of It.”**

Gartner Blog Network, 22 February 2013



**In an elastic environment, a series of alerts from a systems manager on host unavailability may be pure noise, due to a normal scale-down during low traffic periods, or because the service can handle individual node failures. Despite the ill fit for monitoring cloud environments, traditional monitoring remains one of the largest categories of spend in the systems management space.**

Although element managers are able to send events and generate alerts when individual hosts encounter errors, they weren't built for a service-wide view of the patterns and trends determining performance. Without analytics that aggregate metrics and provide a more dynamic view of performance relative to meaningful thresholds, even Manager-of-Managers systems are only monitoring at the surface of any environment. They don't address the service-level monitoring required to operate more sophisticated architectures made up of open-source stateful services, message buses, containers, and orchestration tools in the cloud.

*Despite the ill fit for monitoring cloud environments, traditional monitoring remains one of the largest categories of spend in the systems management space.*

# Waiting for the Next Health Check

## Infrastructure Components



Hardware



Cloud Services



Web Servers



Operating Systems



Virtual Machines



Containers



Orchestration Tools



Databases



Message Buses



Stateful Services



Networks



Custom Applications



## Element Managers



Component Managers



Event Monitoring



Alerting Tools



App Performance



## Waiting to Pull Data



Delay Hours or Days



## Managers-of-Managers



System Managers



Alert Consolidation



## Operations



Analysis



Decision-Making



Triage Solution



Take Action



Monitoring Maintenance



Rate of Change



Percentile



Absolute



Mean Average



Variance



Integrate



LN / Log10



Square Root



Scale



Sum



Standard Deviation



Count

## The New Way: Metrics Aggregation & Intelligent Alerting

**Analytics on time series data underlies a modern approach to infrastructure monitoring and is key to ensuring availability of today's distributed, elastic environments in production. Analytics help aggregate service-level metrics for a better way to explore performance than a component view alone.**

Rather than simply waiting to pull simple events or consolidate and analyze alerts from a variety of noisy element managers (as alert aggregation tools do), a more effective solution requires real-time alerts on the metrics that actually matter to your specific architecture. By computing and visualizing rates of change, percentiles, moving averages, or variance relative to historical benchmarks, you can isolate a pattern, measure its severity, and correlate the root cause with the trend you're observing to prevent an issue before it affects availability.

By aggregating metrics and comparing against dynamic thresholds (rather than the static limits used by element managers), you can troubleshoot and triage problems at any level of the stack in real time. Dynamic thresholds allow you to compare metrics against a chosen benchmark that may change over time—for example, the historical norm for a given time of day and day of week. The ability to spot and fix even a subtle change in latency, load, or throughput as it emerges is key to proactively operating modern applications in the cloud. For the first time, you can determine the difference between a normal change, an anomaly, and a threatening pattern to get alerts and address issues before they turn into emergencies and affect the end-user experience.

Infrastructure monitoring built on analytics also helps eliminate the false-alarms and alert fatigue that can result from simplistic health checks. By using a push model, where metrics and their corresponding metadata are reported at a regular cadence to an analytics system, an administrator can build an alert that's based on a dynamic query (e.g., alert any time a machine reporting itself as part of the login service has a CPU anomaly). Unlike other monitoring and management tools that require reconfiguration every time you change your environment, charts and alert rules created through dynamic queries automatically survive any and all updates.

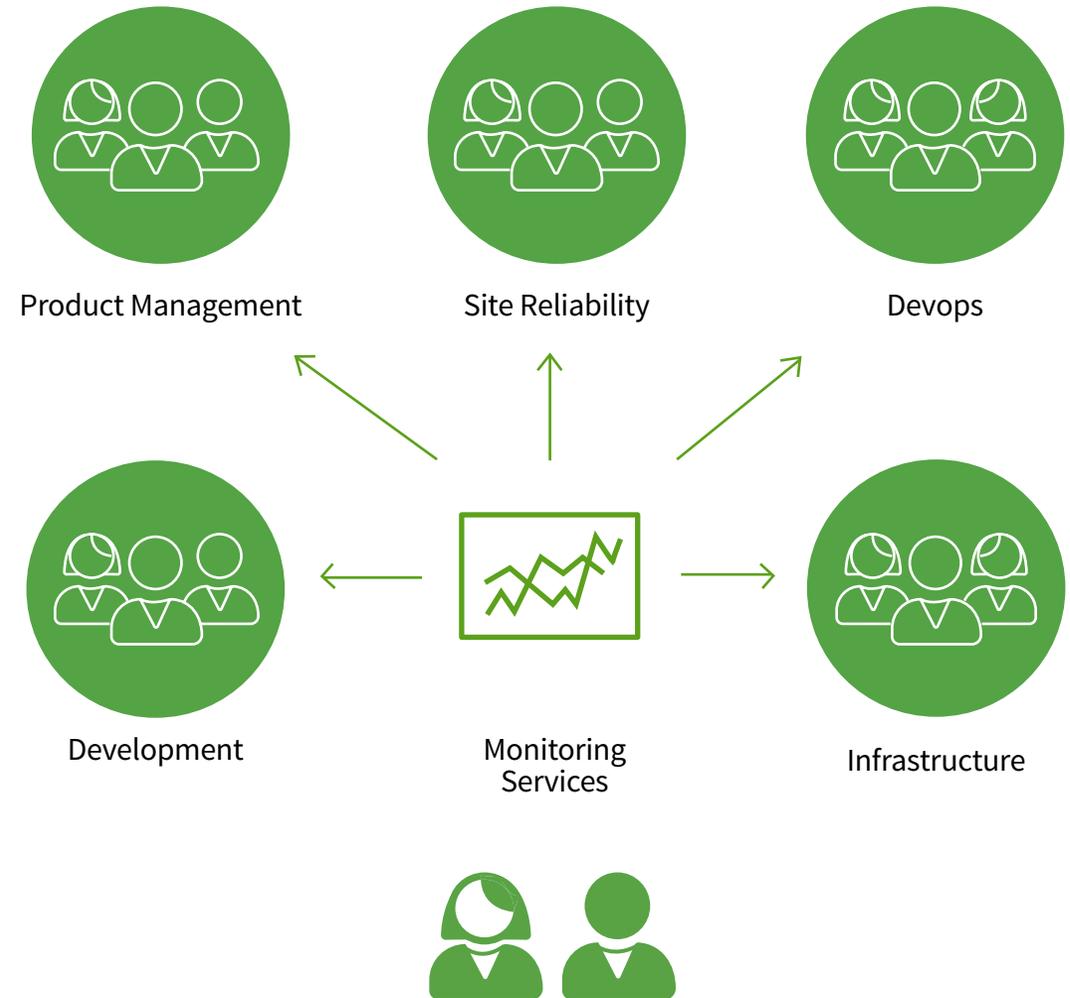


*For the first time, you can determine between a normal change, an anomaly, and a threatening pattern in production before the end-user experiences a difference.*



With the real-time insight introduced by modern infrastructure monitoring, application developers, infrastructure engineers, and operations teams can collaborate across the entire application lifecycle for the first time.

Infrastructure monitoring complements services like application performance management (APM) and log management by filling a large gap not previously addressed: intelligent and timely alerting on service-wide issues and trends within your production environment.



Specifically, developers use an APM solution like New Relic or AppDynamics to instrument their applications and trace performance issues across transactions. However, APM data represents just one subset of information that a modern approach to infrastructure monitoring needs to process. By combining data from APM and several other element managers, a modern infrastructure monitoring solution can aggregate and alert on the metrics flowing directly from the constantly changing population that makes up most elastic, distributed architectures.

To evaluate an issue in production, log management tools like Splunk and the Elastic Stack help operations teams explore all the details of an event and determine root cause after-the-fact. But the massive detail that logs provide can't realistically be processed quickly enough to deliver the meaningful, proactive, and timely alerts that are required to operate today's distributed, scale-out environments.

**A complete development and operations work flow requires real-time alerts that are triggered by the metrics you care about, aggregated at the service level. For every cloud application, infrastructure monitoring focused on time series analytics is essential to availability across the product lifecycle.**



“Health checks didn’t provide aggregations like percentiles or enough metric resolution to make timely decisions and catch problems. [We use an APM] for stack traces... but monitor AWS, Elasticsearch, Kafka, Zookeeper, and Cassandra with SignalFx.”

**Stan Chan**

Head of Core Infrastructure

# The Monitoring Maturity Spectrum



## Element Managers

Components are evaluated via simplistic IT health checks. Element managers pull status and report events when machines don't respond or behave out of the ordinary, even if service performance is unaffected.

*Examples: Nagios, HP OpenView*



## Manager-of-Managers (MoM)

Correlate the abundance of operations data from element managers. But without analytics to aggregate metrics and alert on dynamic thresholds, MoMs fail in complex, elastic cloud environments.

*Examples: IBM Tivoli, BMC PATROL, CA Unicenter*



## Application Performance Management (APM)

Developers trace code performance across app transactions. APMs were not designed for service-level monitoring and alerting, where infrastructure interactions and production factors can create real issues.

*Examples: New Relic, AppDynamics*



## Log Management

Operations teams explore all the details of an event and determine root cause of an issue after-the-fact. But massive log detail can't be processed quickly for proactive alerts on a live scale-out environment.

*Examples: Splunk, Elastic Stack*



## Modern Infrastructure Monitoring

Time series analytics underlie a modern approach to infrastructure monitoring. Performance of production cloud environments require real-time metrics aggregation and alerting on service-wide issues and trends.

*Example: SignalFx*

LEAST ACTIONABLE INSIGHTS

MOST ACTIONABLE INSIGHTS



“Setting up a production threshold in our APM vendor’s tool, we would get 50 emails for a single alert.

We just constantly had alert fatigue from false positives.”

**Florian Berckemeyer**  
Manager of DevOps

## APM Is Not Infrastructure Monitoring

The primary objective of APM is to test pre-deployed code against downstream performance issues. Performance engineering with APM allows developers to deploy an agent that simulates the various transactions performed in the execution of code in production.

By tracing through all the steps across the application stack for a single coding language, the team can approximate the time required to complete API calls and component behaviors against a battery of web, mobile, and desktop scenarios. Developers are then able to detect operational problems, bottlenecks, or inconsistencies prior to pushing the code to live.

**APM solutions should be used for what they are exceptional at doing: providing transaction traces and identifying bottlenecks in code. They were not designed for monitoring the service-level operations of today’s diverse environments, where several factors outside of your code can create real issues.**

Although many APM solutions now come bundled with some basic infrastructure monitoring, they lack the breadth of coverage and context to provide adequate alerting in a heterogeneous production environment. Did you experience high latency between two services because the network was slow or because a load balancer was misconfigured? Was there an unusually high amount of load on that service to begin with? Were several of the nodes in that service down, and capacity was degraded?



VS



#### APM is GOOD for:

- ✓ transaction traces
- ✓ finding code bottlenecks
- ✓ performance engineering

#### APM was NOT designed for:

- ✗ service-level monitoring
- ✗ analytics
- ✗ intelligent alerting



*With intelligent alerting, get notified of changes that are relevant to the overall health of your systems as they happen and act on emerging trends before availability becomes an issue and performance suffers.*

Moreover, most APM solutions require proprietary agents that perform byte-code injection. Though such a heavyweight approach might be acceptable in a development environment, most organizations prefer not to endure the expense of running a proprietary agent across the production fleet and choose to sample data from selected nodes for infrastructure monitoring instead. However, sampling doesn't provide a reliable view of the production environment's changing population or specific performance and is, therefore, an insufficient source of content to drive effective alerts.

APM tools help organizations easily instrument and identify bottlenecks in their code. APM vendors focus most of their development resources on the instrumentation part of the problem (e.g., providing the best tracing for Java applications), but have not invested in the downstream analytics, correlation, and alerting required of a general-purpose monitoring solution. Ultimately, they provide another source of insight that is tremendously valuable when combined with other operational data in a complete, modern infrastructure monitoring solution.

# Log Management Still Needs Infrastructure Monitoring

The immense volume of log data generated by modern infrastructure offers operations teams deep insight into the root cause of a systems problem. Logs are primarily unstructured data, typically in the form of message streams, that are emitted from applications as a detailed record of events. By auditing and exploring log data in the context of the application that created it, engineers can troubleshoot code or system bugs for deep evaluation of time-sequenced issues.

Analysts can also use logs to enrich other data sets to gain intelligence into machine and user interactions. Log data is used not only for server, network, and software troubleshooting, but also for regulatory and security compliance, forensics, and incident investigation.

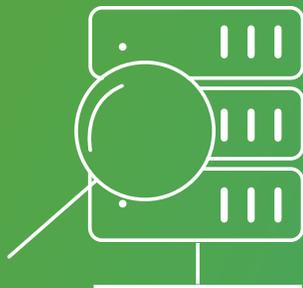
**However, logs are not particularly useful for alerting on real-time infrastructure issues across distributed environments. At the time of an emergency, an infrastructure monitoring solution provides the necessary service-level details to triage and remediate the issue.**



“Before, we didn’t have a way to go from a metric in the app code to an alert. Now, everyone in dev and ops knows how to create a meaningful alert for themselves.”

**Mohit Gupta**

Product and Engineering Lead for Infrastructure



Log Management is

**GOOD** for:

- ✓ batch analytics
- ✓ post-event analysis
- ✓ compliance

Log Management was

**NOT** designed for:

- ✗ real-time
- ✗ metrics aggregation
- ✗ intelligent alerting

Metrics are the best first line of defense when dealing with a problem. Streamed into an analytics-based monitoring solution, they help the viewer narrow down to the service and application causing problems in the most timely manner. Even more effectively, modern infrastructure monitoring can generate proactive alerts on patterns that foretell a mounting concern and provide the runway to isolate, assess, and address the underlying issue before a problem affects the end user.

Because logs are primarily unstructured data, they are well suited to batch data analysis of a discrete event. However, a big data approach to logs makes them poorly suited to the real-time search and stream processing required for timely alerts. The high volumes of disk I/O and network load needed for log exploration are much better aligned to post-hoc analysis, as opposed to the high metric throughput typical of a time series database used for infrastructure monitoring.

For cloud environments, whose goal is to scale infrastructure elastically, you need a purpose-built system focused on metrics and analytics. Real-time aggregation is a job not fit for batch analytics because alerting requires much faster, more flexible insights. Log analysis for deeper exploration and investigation is ultimately a great complement to an infrastructure monitoring solution that handles real-time analytics and alerting on time series data.





“Health checks create so much noise. We transitioned to metrics-based monitoring to alert on aggregations.

[We use an APM], but only for code-level stack traces, [not our AWS production environment].”

**Weston Jossey**  
Head of Operations

## APM + SignalFx + Logs

**Today, a more modern approach to infrastructure monitoring can help rationalize the role of the APM and log management tools that development and operations teams already use to understand the lifecycle of their applications. The data and insights at each stage of the journey shouldn't be viewed in three separate silos. Today's smartest product organizations are managing both effectiveness and cost by flowing insights across all stages of the application lifecycle.**

Using the metaphor of an airplane, APM serves as the essential pre-flight testing, infrastructure monitoring provides in-flight systems intelligence, and log management is the black box recorder for deeper post-flight analysis.

# Application Lifecycle Management

APM

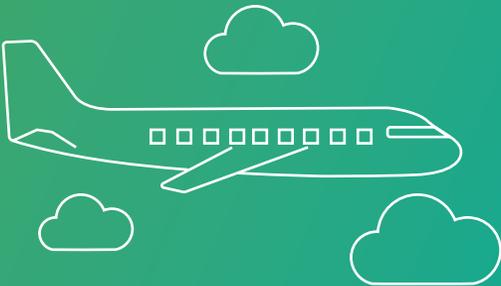
Performance Testing Pre-Flight



⌚ Luxury of Time

signal fx

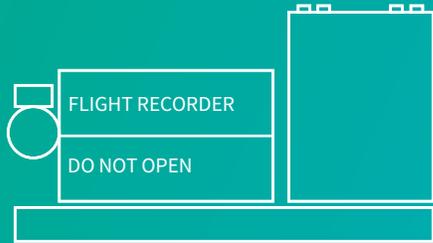
Streaming Metrics Aggregated In-Flight



✓ Real Time Matters

LOG MANAGEMENT

Black Box Recorder Post-Flight



⌚ Luxury of Time

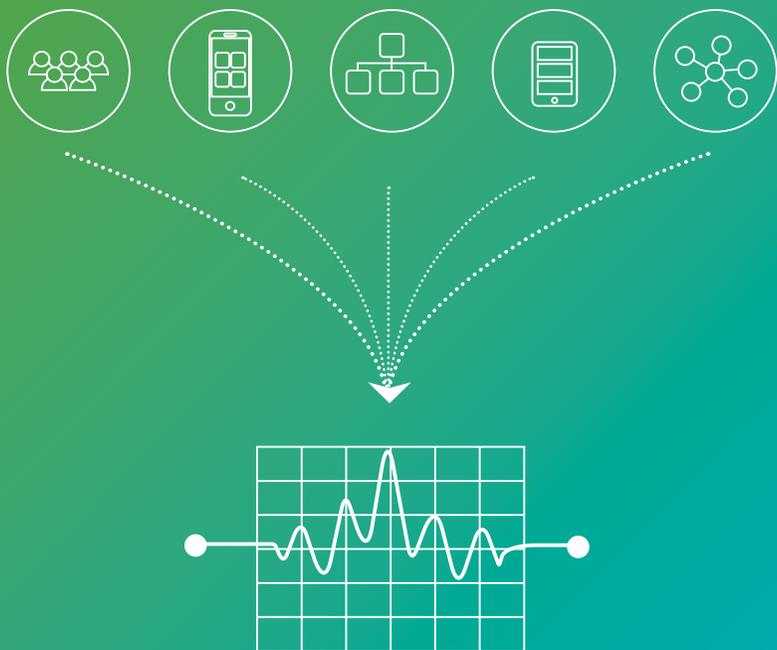
An APM solution provides the same code-level insights through transaction traces that the battery of pre-flight tests evaluate relative to expected passenger number, freight weight, weather conditions, fuel storage, and flight distance. Like a jet engine, performance engineering prior to deploying code tests how the existing network and architecture are likely to interact with code changes downstream.

Just as conditions can change rapidly in response to the unpredictable nature of flight, streaming analytics on infrastructure and application telemetry help ensure that the operations team is focused on the right things at any given moment. During the flight, like in a distributed cloud environment in production, alerting on the atmospheric and computer data that indicate performance relative to historical and expected patterns is the best way to know when a pilot (or infrastructure engineer) needs to take action. A modern infrastructure monitoring solution aggregates metrics from every element manager, including APM, to become the system of record for real-time operations.



*A modern infrastructure monitoring solution aggregates metrics from every element manager, including APM, to become the system of record for real-time operations.*





*SignalFx is the best way to aggregate and alert on streaming metrics, filling the monitoring gap between APM and log management.*

After the plane has landed, and you once again have the luxury of time, log management helps engineers evaluate any systems errors in-depth to ensure that specific action can happen prior to the next scheduled flight to avoid similar events. Like the black box recorder, logs are incredibly useful for root cause analysis after a problem has occurred. Once the outcome is known, log management provides many of the details on chains of events, dependencies, and the decisions that contributed along the way.

SignalFx is the best way to aggregate and alert on streaming metrics, helping today's dev and ops teams fill the gap between APM's pre-flight performance engineering and log management's post-mortem event analysis. SignalFx's real-time visibility into and analytics on the live production environment also help rationalize your existing investments with better overall results.

# What's Your Monitoring Strategy?

*Start Building an Application Lifecycle Management Engine*

**Learn More About  
Modern Infrastructure  
Monitoring**

**TALK TO US**

**Get Started with a  
Free 14-Day Trial**

**FREE TRIAL**

**Check Out a  
Video Tutorial**

**WATCH NOW**