chronosphere

4 ways metrics speed up troubleshooting and repair

Metrics, as the numbers representing data measured over time intervals, help organizations understand how systems are supposed to behave. When systems stop working properly, metrics are also the most effective tool in helping technical teams get them back up and running quickly, according to the O'Reilly **Cloud Native Monitoring Report**.

Why lean into metrics?

Although multiple signals—logs, traces, and metrics—at times may be required to solve a very difficult problem, starting with metrics is always the most logical. The reason is because metrics enable teams to move from the broadest view down to the narrowest.

Here are four things teams gain from a metrics-first approach:

1. **An efficient snapshot of the system**
2. **Ease of use and implementation**
3. **Fast data aggregation and comparison**
4. **Ability to create alerts**

If your team is looking for faster system troubleshooting and remediation, here's why starting with metrics from your observability platform makes the most sense.

1 Metrics provide an efficient snapshot of the system

Metrics efficiently allow your technical team to understand the current state of a system. Together with the contextual knowledge of your organization, teams can ask questions and prove or disprove assumptions and start troubleshooting right away.

If you want to know if there is an *HTTP : 503 Service Unavailable* error and when it happened, for example, you can use context and metrics to get answers fast—without a JSON filter or a full-text search for all HTTP response codes. Prometheus scrapes metrics HTTP end-points on monitored targets so if you have an HTTP server that's producing an *HTTP : 503 Service Unavailable* error, it's easy to detect that quickly by graphing a metric in Prometheus.

These four types of metrics are generally available from Prometheus and other platforms:

- **Counters** – These are cumulative metrics that can only increase (or not change). Examples include distinctly numbered things such as HTTP requests, RPC calls, or even business metrics such as number of sales.
- **Gauges** – These are singular metrics that can either increase or decrease. Examples include temperature, speed, or memory usage; even metrics that decrease, such as concurrent HTTP requests.
- **Histograms and summaries** – These are sample observation metrics. Examples include request duration, request sizes, or ranking. They are similar, but histogram is best for aggregation, since teams can use quantiles.

Common metrics types

- Counters
- Gauges
- Histograms
- Summaries

2 Metrics are easy to use and implement

Metrics deliver the most value to your technical team and provide the most utility, with the least amount of set up effort. Moreover, the open source tools teams are already using are likely to support Prometheus exposition formats. In NGINX, PostgreSQL, and Kubernetes, for example, user communities have built Prometheus exporters that you can run using just a few lines of configuration.

Querying metrics is a very efficient way to generate the alerts you need before you even go to individual logs.

3 Metrics let teams aggregate data quickly and compare it

When you continually collect metrics, you have a way for your organization to get information quickly and observe aggregated data over time, which can be a significant advantage in preventing downtime. Although there are multiple ways to aggregate metrics, depending on the type, metrics are simply the fastest way to combine data to troubleshoot and fix a system.

Using metrics, teams can count bounces in the last hour or the last day; see where the most bounces occur; and at which endpoints.

4 Metrics let you create alerts

As the name suggests, alerts let you know when there's a problem. They're meant to tell you when the system reaches a certain threshold that you've set, such as a number of HTTP requests or a certain error being generated. Querying metrics is a very efficient way to generate the alerts you need before you even go to individual logs.

Why metrics? Troubleshoot systems faster, prevent downtime.

Metrics, which have proven to be easy to use and implement, provide low-latency impact analysis for data comparison over time. That makes them an ideal first line of defense in advancing rapid resolutions to real-world system downtime occurrences.

Download the Cloud Native Monitoring Report from O'Reilly to learn more.

Download

Learn more and request a demo at **chronosphere.io**

chronosphere