The need to introspect and understand systems and services is not new — many of the basic goals of observability have been in practice for decades. What’s changed is the nature of the applications and infrastructure that teams are operating. Cloud-native applications running on containers and microservices have a completely different architecture and are designed to be more scalable, reliable and flexible than legacy apps. Cloud-native has forced organizations to revisit how they perform monitoring and observability because:

- **Data is growing in scale and cardinality.** Cloud-native environments emit a massive amount of observability data — somewhere between 10x and 100x more than traditional VM-based environments.
- **Systems are more flexible and ephemeral.** Both the usage patterns and retention requirements are vastly different to what they were pre-cloud-native.
- **Services and systems have greater interdependencies.** Breaking services down into microservices leads to more complex dependencies that engineers must understand in order to troubleshoot problems. This also results in a greater need to correlate and connect infrastructure to applications to business metrics.

All of this has led to an explosion in complexity that makes it nearly impossible to reliably and efficiently operate cloud-native services without dramatically increasing overhead or finding a new approach.

On top of the technical complexity, there is increasingly a layer of organizational complexity that exacerbates today’s challenges. There are now more distributed teams creating more dashboards, reports, and alerts, driving increases in metadata. Central observability and site reliability teams are left to try to decipher a mounting pile of data.

Data is growing in scale and cardinality. Cloud-native environments emit a massive amount of observability data — somewhere between 10x and 100x more than traditional VM-based environments.

Systems are more flexible and ephemeral. Both the usage patterns and retention requirements are vastly different to what they were pre-cloud-native.

Services and systems have greater interdependencies. Breaking services down into microservices leads to more complex dependencies that engineers must understand in order to troubleshoot problems. This also results in a greater need to correlate and connect infrastructure to applications to business metrics.

Open-source solutions are time consuming and unreliable as you scale

Many companies find that once they reach a certain scale, they begin to struggle with the reliability and complexity of their Prometheus setup. The result is that you often end up flying blind because metrics are not available or are too hard to find quickly. Teams also end up spending a lot more time managing Prometheus and Jaeger, when they could be focused on other things.

SaaS solutions struggle to keep up with cloud-native

As organizations transition to cloud-native infrastructure, they often find that their existing SaaS observability tools are not keeping pace. The cost of the solution is outpacing the value they are delivering. They are unreliable, AND they force teams to make tough decisions about what data to keep, which ends up being very time consuming.

**Our Solution**

Chronosphere is the only observability platform that puts you back in control by taming rampant data growth and cloud-native complexity, delivering increased business confidence. Engineering organizations at startups to well-known global brands in the Fortune 500 around the world trust Chronosphere to help them operate scalable, highly available, and resilient applications.
A few ways we’re different from other monitoring solutions out there:

**Keep cost under control as you scale**
Chronosphere enables you to keep observability costs from increasing exponentially, even as you grow to thousands of engineers and millions of data points.

**Future proofed Observability**
Chronosphere is powered by open source and allows you to leverage existing Prometheus AlertManager, and OpenTelemetry investments for greater speed and flexibility.

**Scale reliably as you transition to cloud-native**
Chronosphere was built from the ground-up for cloud-native scale and complexity, enabling us to provide greater reliability than any other SaaS offering on the market.

**Engineers can solve problems faster**
Chronosphere helps engineers be more productive by giving them faster and more actionable alerts that they can triage rapidly, and allows them to spend less time on observability instrumentation.

How it works

**Step 1.** Deploy the Chronosphere collector. The Chronosphere collector ingests metrics and traces and sends them to the Chronosphere backend. The collector is compliant with all major open source standards — Prometheus, OpenTelemetry, and older formats like Graphite/StatsD.

**Step 2.** Set thresholds in the Chronosphere control plane. Determine how metrics and traces should be aggregated, dropped, or rolled-up based on environment and business need. Tune ingest as needed to control for cardinality spikes. Set your retention by application or environment.

**Step 3.** Run queries, create dashboards, set alerts. Let teams use the open source formats they already know and love: Grafana for dashboarding/visualization, Prometheus Alertmanager for alerts, and PromQL for queries.

**Step 4.** Take back control. With an end-to-end managed SaaS offering, you’ll not only save engineering time, but will gain a solution that is more reliable than self hosted open-source tools and any other observability SaaS offering. On top of that, Chronosphere’s unique pricing model puts you in control of the bill, meaning you’ll never see an overage again.

Ready to learn more? Book your demo today by visiting chronosphere.io