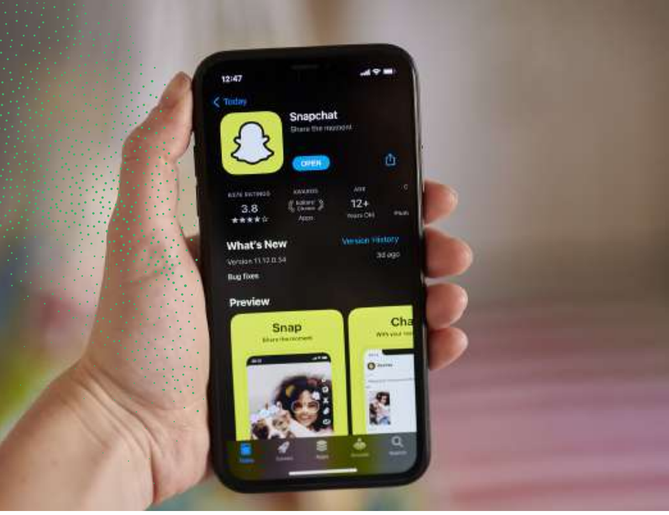
chronosphere

CASE STUDY



# How Snap increased observability reliability and improved developer experience

Snap Inc.

## Executive summary

Launched in 2011, Snap, Inc. is a technology company that serves over 750 million daily customers worldwide with Snapchat, Spectacles, and Bitmoji. These products let consumers share experiences with friends, use VR/AR, and develop a social network avatar.

As they scaled, Snap's in-house, open source observability solution was getting expensive and time-consuming for engineers to realistically manage as the company saw massive user growth.

## The challenge

In order to be competitive in a crowded market and delight the millions of customers they serve, Snap must deliver best-in-class availability and performance to customers worldwide. However, the company's previous observability setup wasn't meeting expectations in terms of:

- **Availability**. The legacy Graphite system faced stability issues and was constantly failing. Every time the system crashed, engineers had to manually step in and bring systems back online – which was time-consuming and costly. Furthermore, the system had performance issues which meant that dashboards and queries would load slowly, or not at all, meaning engineers couldn't respond quickly to customer-facing issues.
- **Scalability**. Snap's observability system was at its limits and couldn't keep up with the amount of daily ingested data. This meant that Snap was only able to ingest and retain a fraction of the metrics they really needed for troubleshooting purposes. As Snap moved to a cloud native architecture, they knew they would need a system that could provide all the critical metrics at a reasonable cost.

- **Usability**. Snap found that their self-managed Graphite instances were difficult to use, time consuming to manage, and negatively affected developer experience. It was also costly to run in terms of infrastructure costs and people hours.

## Solution

Snap has been a long-time partner to Chronosphere. Evan Yin, technical lead at Snap, says he discovered the technology when it was still a GitHub page for M3 – before Chronosphere's official founding – as part of his research for a new observability solution. Two key goals for Snap for their new solution would be to improve developer productivity and cost efficiency.

According to Yin, Snap chose Chronosphere, because of its ability to control observability data volumes, provide high levels of availability at scale, as well as being suited for cloud native architectures (including Prometheus support). He also felt that his values about observability aligned well with the Chronosphere founder's vision for the company, which provided a favorable foundation for such a partnership.

"Chronosphere is built to specifically address issues in the cloud native world. We can always rely on them to solve the problem," he says.

### Reduced costs

For observability at scale, working with Chronosphere reduced data volumes by more than half and saved thousands of engineering hours. Using Chronosphere's Control Plane, the team defines what data labels are most important and which are noise, making it faster and easier to triage issues while significantly reducing costs.

### Support for cloud native architecture at scale

As part of its infrastructure upgrade, Snap adopted Google Kubernetes Engine (GKE), due to its customization capabilities and flexibility. Chronosphere is built to oversee cloud native architecture and easily integrates with Google Cloud offerings. Due to Chronosphere's ability to support large scale environments, Snap was able to scale their observability solution 5x, from 50 million time series to 250+million time series to run all of the use cases they needed.

## Improved observability reliability

Chronosphere offers an industry-leading 99.9% uptime SLA that has never been broken. Compared to the challenges with downtime and reliability that Snap experienced with their previous solution, this was a big improvement.

“Our developers now have the freedom to emit high cardinality metrics and load dashboards faster. They do not have to worry about metrics availability anymore,” said Yin.

## Developer productivity

Since moving to Chronosphere, the central observability team has seen a 90% decrease in on-call pages. The team can now work on value-added tools for Snap – instead of just trying to constantly put out fires in the observability platform. On top of that, the rest of the Snap engineering team can be more productive with faster loading dashboards and queries and less time spent worrying about metrics.

# Outcome

Snap’s adoption of Chronosphere not only helped the company set itself up for future success; it also ensures application availability for its millions of daily users and made behind-the-scenes operations much smoother and more reliable.

“We’re trying to provide the best in class observability tools for the service owners at Snap so they can manage their services more smoothly and efficiently,” says Yin.

“At Snap we’re religious about cost efficiencies, so when we built our observability system we asked ourselves: how can we reduce waste right from the beginning?”

## Key results

As an engineering team, Snap is experiencing with Chronosphere:

- Greater uptime of the observability stack: The team now has 99.9% availability and has yet to miss an SLA since implementation.
- Reduced costs: Using the control plane, Snap has reduced data volumes by more than 50%, leading to significant cost savings.
- Fewer on-call pages: Snap has cut down the number of on-call pages the central observability team receives by more than 90%.
- Improved performance: With the combination of Google Cloud and Chronosphere, Snap's developers can now rely on their dashboards to load and view high-cardinality metrics instead of regularly having a system crash.
- Improved developer experience and efficiency: Chronosphere has saved thousands of engineering hours and drastically improved developer experience, so they can have more time to work on value-add features for the company.

> "We always want our developers to be able to deliver the features faster or even just load their dashboards faster so that they can resolve the incident faster".