![chronosphere]

CASE STUDY

It's all about scale: Why DoorDash needed true cloud-native monitoring

Chronosphere provides a scalable, reliable and customizable monitoring-as-a-service purpose built for cloud-native applications

DOORDASH

The Challenge: constant metrics data loss

At first, DoorDash was using a combination of StatsD monitoring for its cloud-native stack and another solution to monitor its virtual machine environment. As the **cloud-native environment scaled** and developers delivered new features, however, the monitoring system kept breaking down.

"We were experiencing constant packet loss," explained the observability lead. "Developers could break the whole system by making some benign change that has some weird bug and it crashes the whole system." There was an extreme noisy neighbor problem — any change made by developers could easily impact StatsD's ability to monitor some other, seemingly unrelated application in an entirely unpredictable way.

It was so bad, he said, that when he first joined the company no one was actually paying attention to the metrics generated by StatsD, instead relying on work-arounds like counting log lines. When the team

upgraded StatsD, they discovered that they had been losing metrics because of a bug in the system.

"We did the upgrade, and there was an instant change in the pattern," the observability lead said. "That's the bigger problem: The pattern changed. That means we had been flying blind for a long time. We had been seeing a zigzag, but then it became a flat line, which was more or less what we expected." The team had been suspecting that something was off, but hadn't found a way to verify it.

In general, DoorDash thinks of itself as being data driven. Everyone is crazy about numbers, from the CEO down to the newest engineer. Data is used to make better decisions about technology and about the business. If everyone loses observability, it means the entire company loses that competitive edge. Because software is a core part of DoorDash's product, losing visibility into the application suite was simply not acceptable.

Open source

A solution that was based on open source technology was really important because of the team's experience with closed-source solutions in the past. The proprietary format made it difficult for engineers to learn how to use the system and meant that any customization was essentially impossible. Ideally, the solution would have a minimal learning curve and build on technology most engineers were already familiar with.

Scalable

The new solution needed to be able to scale without losing data and without becoming extraordinarily expensive. DoorDash already had a massive StatsD cluster and was experiencing timeouts because it was unable to control the incoming data traffic. It felt like the current solution was at the limits of what it could handle, scale-wise — and yet DoorDash intends to keep growing.

Looking for a solution

The DoorDash team required a solution that was:

- Open source
- Scalable
- Reliable
- Fully distributed

Reliable

Given the problems with data loss, reliability was key. DoorDash was looking for something that developers couldn't break with a seemingly innocent code change and that didn't buckle when asked to scale. Relatedly, eliminating the noisy neighbor problem was important, so that the monitoring system wouldn't experience cascading outages.

Fully distributed

When you're operating at scale, central operations become a bottleneck, explained the observability lead. Dealing with hundreds of millions of datapoints per second could overload the endpoint processes. In his view, the key to being able to scale was having a fully distributed monitoring system. This distributes the load, obviously, and limits the impact of any individual failure. In other words, a distributed system was the only way the observability lead thought that a monitoring system could meet DoorDash's scale and reliability requirements.

Success with Chronosphere

"We don't even discuss metric loss," the observability lead said, about the difference between using StatsD and Chronosphere.

"Of course, we are always going to monitor our monitoring. But we have a lot more peace of mind now."

In general, engineers are able to set and forget the **cloud monitoring tool** because they know it is working. Chronosphere's much simpler metrics pipeline has also reduced not just packet loss but all kinds of other operational issues.

Moving to Chronosphere has also underscored just how ad-hoc the company's tagging system had been, and now they are working on a much more unified, consistent naming and tagging practice.

Perhaps most importantly, DoorDash is no longer flying blind. "We have metrics, and we are using them," the observability lead said. DoorDash doesn't need anything fancy when it comes to monitoring, he said — just a solution with a strong foundation that provides the scalability and reliability they need.
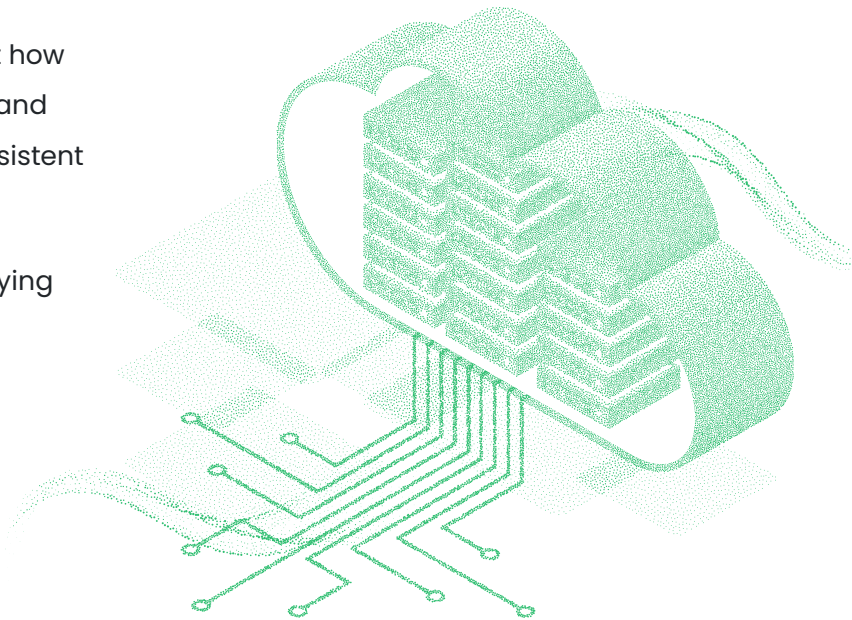
"The visibility and control that Chronosphere's platform gives us to manage our observability data and costs are a game changer, especially with our unprecedented growth."

– *Ryan Sokol, VP of Engineering*